SFU

TRIWAVE SYSTEMS

ENSC 440

Akriveia Beacon

Test Plan

Team 5

21/10/2019

Project Team

Jeffery Yeung CCO
Keith Leung CTO
Scott Checko COO
Ryne Waterson CIO
Jerry Liu CEO

Contact

Jeffery Yeung
zjyeung@sfu.ca

Submitted to

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University

Issue Date

October 21, 2019

Contents

1 Introduction

The following document contains the detailed Test Plan for Akriveia Beacon - The Indoor Location Rescue System created by TRIWAVE SYSTEMS. The Akriveia Beacon system focuses on improving the location and rescue process of personnel trapped in buildings during and after small scale disasters such as fires and low magnitude earthquakes.

Such a system allows for search and rescue operations to be carried out more safely and reliably. The core functionality allows for trapped victims to be located in a disaster situation confined within complex urban environments. By pinpointing the exact location of any victim wearing an ID tag, the search and rescue time for first responders is minimized; which is crucial in any disaster rescue operations. This Is achievable by incorporating combinations of advanced Ultra-wide-band radio modules and microcontrollers to create a dependable indoor positioning system using trilateration methods.

This test plan is divided into four main sections: system, hardware, software, and electrical tests. Each test plan section will detail the testing procedure and provide comprehensive test cases derived from design requirements that directly validate primary and secondary functionality of each of the major components and the overall system of Akriveia Beacon. Furthermore, this document will outline major functionalities and features and show the quantifiable, observable, and realistic factors that are incorporated into the design of the system. All test cases for each of the four test plans follows the below format.

[**TEST-#**]

Code	Definition
TEST	Test Plan abbreviation.
#	Test Case ID

Table 1: Test Case Encoding

Each test case is grouped into four sections shown below and their IDs correspond with a letter representing the system component.

Component	Abbreviation Code
System Level	SL
Hardware	HW
Software	SW
Electrical	EC
Usability	US

Table 2: Planning Stage Abbreviation Code

Testing equipment and facility needed are provided by SFU and are available for the test team to use and perform. TRIWAVE SYSTEMS is dedicated to create a reliable and robust indoor tracking system, designed to improve disaster search and rescue operations with human safety as the pivotal focus.

2 System Level Test Plan

The Akriveia Beacon consists of various hardware, electrical and software components. As a system that will be operating under environments that dangerous during emergency disasters, the system must be tested rigorously to ensure minimal error or failure rate. Each major components must be tested throughput individually and together as an integrated system. Below are test cases detailing the steps and procedures of conducting system level tests for the Akriveia beacon to ensure that its main functionalities are working as expected and meets specified requirements.

ID	Test Procedure	Validation
SL-01	Send start tracking command to beacon system from web server UI. ID tag is turned on into tracking mode.	All active beacons receive and return acknowledge back to server. Tags are ranged with correct distance values received by server from all beacons.
SL-02	Send end tracking command to beacon system from web server user interface. ID tag is turned off.	All active beacons receive and return acknowledge back to server. Server stops receiving tag ranging values from beacons.
SL-03	System is set in tracking mode (steps in SL-01). Tag ranging is verified by physically measuring distance between tag and beacon with tape measurer	Compare physical measured distance to beacon ranging values received on web server. The difference must not exceed 0.5m as specified in the design requirements.
SL-04	System is set in tracking mode (steps in SL-01). ID tag is turned on into tracking mode. Tester must wear an ID tag and be able to walk freely in test area to validate trilateration algorithm.	Verify Trilateration method is implemented properly. Tracking resulting in correct coordinates when personnel wearing id tag is moving around the test area. Difference in coordinates must not exceed 0.5m.
SL-05	System is set in tracking mode (steps in SL-01). Multiple ID tags (at least 2) are present in the test area and are turned on in tracking mode.	Web server receives proper beacon messages. Each individual id tag can be ranged correctly and trilateration calculation yield result with tolerance less than 0.5m.
SL-06	System is set in tracking mode (steps in SL-01). At least 4 beacons must be present and registered in the web server. At least 1 ID tag is present in the test area. During tracking mode power off one of the beacons to simulate beacon failure.	Before beacon failure tracking shows correct results. After beacon failure web server shows which beacon(s) is down and tracking should still return correct ranging results if there are a least 3 beacons operational.
SL-07	Web server is started on server machine hosted on public IP and local IP and is accessible locally and remotely from another device (laptop, tablet, etc.)	System configuration can be performed correctly. Operations such as adding/editing/removing beacons, users, maps through the web server user interface can be performed with correct results

Table 3: System Level Test Cases

3 Hardware Test Plan

The system consists of two major hardware components, the anchor beacons, and the wearable ID tags. Each beacon contains a DWM1000 Ultra-wideband (UWB) transceiver module, a 3.3V 8MHz Arduino Pro Mini (PM33), and an ESP32 WiFi module, and each ID tag contains a DWM1000 module and an Arduino Pro Mini. Hardware components must be tested individually to verify their primary functions and performance.

ID	Test Procedure	Validation
HW-01	DWM1000 UWB to PM33 connections are valid with Serial Peripheral Interface (SPI). Connect Beacon to secondary PC via serial connection.	Reset PM33 by pushing toggling the reset button on beacon. Serial connection should show device initiation message with correct device ID: Device ID: DECA
HW-02	Configure 1 beacon and 1 ID tag in ranging mode. Set 2 device 1m apart measured by tape measure. Start ranging and observe results using serial connection from beacon.	Ranging values can be observed from serial connection on beacon device. The values are within 0.5m tolerance.
HW-03	Multi-Tag detection ability. Configure 1 beacon and at least 2 ID tags in ranging mode. Set 1 id tag 1m apart from beacon and another 2m apart from beacon measured by tape measure. Start ranging and observe results using serial connection from beacon.	Ranging values can be observed from serial connection on beacon device. The values are within 0.5m tolerance. Ranging messages show 1 id tag with 1m range values and the other id tag with 2m range values. Unique tag id is also matched correctly with each range.
HW-04	Beacon can be started by sending start command via serial. Configure 1 beacon and 1 ID tag in ranging mode.	Connect to beacon using serial connection. On PC connected to beacon via serial, send command start, the beacon returns `start_ack` to acknowledge command and begins receiving range values.
HW-05	Beacon can be stopped by sending end command via serial. Configure 1 beacon and 1 ID tag in ranging mode.	Connect to beacon using serial connection. On PC connected to beacon via serial, send command end, the beacon returns `end_ack` to acknowledge command and stops receiving range values from ID tag.
HW-06	Beacon on/off state is saved in non-volatile memory. Configure 1 beacon and 1 tag in ranging mode. Start ranging with command start. Power cycle beacon.	Before power cycle beacon outputs correct ranging values, after power cycle beacon initialization messages are shown and continues to output ranging values.

Table 4: Hardware Test Cases - Part 1

ID	Test Procedure	Validation
HW-07	ESP32 is configured to connect to WiFi access point and receives IP with pre-specified port. Secondary PC device is also connected to the same network and receives IP. IP on PC is host IP.	Using netcat commands on secondary PC device to send and listen for UDP packets. All packets are sent to ESP32 properly and can be observed via serial connection from ESP32. Listen: nc -ulk 0 <PORT > Send: nc -u <IP ><PORT >
HW-08	Hardware serial between PM33 and ESP32 is established by connecting: ESP32 G17 ->PM33 RXD ESP32 G16 ->PM33 TXD ESP32 GRD ->PM33 GRD ESP32 is connected to secondary PC using USB cables.	Hardware serial communication between PM33 and ESP32 verified by resetting PM33 and observing PM33 reboot message on ESP32 side. Using COM port viewer the reboot message PM33 can be seen from ESP32 serial out on secondary PC
HW-09	Beacon can be pinged by server. Beacon is connected to WiFi access point and web server is configured to connect to the same access point and send ping command.	Server issues ping, beacons receives and sends packet to host IP. Beacon returns packet containing ping acknowledgement and basic device information. Packet should be `ping_ack` , <EUI >

HW-10	Beacon can be rebooted by server. Beacon is connected to WiFi access point and web server is configured to connect to the same access point and send reboot command.	Server issues reboot, beacons receives and sends 2 reboot acknowledgement packets back to server. Packets are:
		`esp_reboot_ack`
		`pm33_reboot_ack`

HW-11	Beacon ranging delay is less than 5 seconds. Configure 1 beacon and 1 ID tag in ranging mode. Set 2 device 1m apart measured by tape measure. Start ranging and observe results using serial connection from beacon.	Observe ranging values from beacon via serial connection with time stamp option enabled. Time difference between each ranging value must be less than 5 seconds.
HW-12	Beacon ranging delay is less than 5 seconds. Configure 3 beacons and 1 ID tag in ranging mode. Set each beacon-tag pair 1m apart measured by tape measure. Start ranging and observe results using individual serial connections from each beacon.	Observe ranging values from beacon via serial connection with time stamp option enabled. Time difference between each ranging value must be less than 15 seconds.

Table 5: Hardware Test Cases - Part 2

ID	Test Procedure	Validation
HW-13	Observe the time for the ESP32 to successfully connect to the configured SSID broadcasted by the dedicated WiFi access point and assigned an IP with configured port. Secondary PC device connected to the ESP32 is used to monitor the setup messages output on COM port viewer.	With time stamp option enabled in the COM port viewer, record the time between the appearance of serial messages "Connecting to WiFi" to "UDP port, <PORT >". Expected time is less than 4 seconds.
HW-14	Default DW1000 PRF config is 16MHz. Analyze ranging improvements by increasing pulse repetition frequencies from 16MHz to 64MHz. Replace `FREQ_16MHZ` to `FREQ_64MHZ` in the PulseFrequency field	Set a beacon and a tag 1m apart measured by tape measure. Observe ranging values from beacon via COM port viewer with PRF at 16MHz, then observe ranging values from beacon via COM port viewer with PRF at 64MHz. Compare their ranging values, the latter setup should yield higher accuracy by 0.1m
HW-15	Observe the time it takes for the PM33 reset and the ESP32 to restart upon issue of the reboot command through serial.	Configure ESP32 to connect to WiFi access point and receive an IP with prespecified port. Connect a secondary PC to the same SSID and assign it the host IP. Using netcat commands on shown in HW-07, issue the reboot command in ranging mode and observe the `pm33_reboot_ack` and `esp_reboot_ack` appear in chronological order. Record the time taken from the issue of the reboot command to when a ranging value first appears again.
HW-16	Observe that the DW1000 device configuration chooses one of the recommended preamble codes as a result of increasing pulse frequency to 64MHz from HW-14 and selecting UWB Channel 5. Based on user manual, recommended preamble codes for 64MHz PRF on Channel 5 are 9, 10, 11, 12.	View serial message output from COM port viewer. In the Device mode line of the device initiation message, PRF should be 64MHz, Channel should be #5 and Preamble code should be 9, 10 ,11 or 12.
HW-17	Calibrate antennas between beacon-tag pairs by finding optimal AntennaDelay value. To find optimal AntennaDelay value, place a beacon-tag pair 5.01m apart, given that both DW1000 are configured with 64MHz PRF and UWB Channel 5. Beacon acts as the initiator while tag acts as the responder.	Monitor the COM port viewer of the responder, record the AntennaDelay value when its associated range value is within the predefined epsilon value of 5.01m.

Table 6: Hardware Test Cases - Part 3

4 Electrical Test Plan

Since beacons and ID tags both consist of microcontrollers, transceivers and various electrical parts, it is crucial that each of these components are powered by proper power supply. In the case where two components are powered by the same battery operating at different voltage and current levels, the system must include regulators that can ensure the proper power is delivered to the components. These regulators must work flawlessly, as an electrical failure in the system could be catastrophic. Furthermore, the hardware components must be tested against short circuit as it can results in an excessive current flowing through the circuit and therefore damaging electrical components and creating system failures.

ID	Test Proceedure	Validation
EC-01	Measure voltage and current supplied to ESP32 power input pin with digital multimeter (DMM).	Observe measured values on the DMM to verify that input from battery is sufficient to power ESP32 and stable.
EC-02	Measure voltage and current supplied to ProMini power input pin with DMM.	Observe measured values on the DMM to verify that input from voltage regulator is between 3.3V and 3.6V to power the ProMini and stable and that input current is 40mA to each digital IO pin.
EC-03	Measure voltage supplied to DWM1000 power input pin with digital DMM.	Observe measured values on the DMM to verify that power input to the DWM1000 is between 3V and 3.5V.
EC-04	Measure logic voltages supplied by the MOSFET reboot circuit with the DMM.	Observe measured values on the DMM to vefify that the 3.3V logic output supplies 3.3V and the 5V logic output supplies 5V.
EC-05	Measure clock frequencies from ESP32 using an oscilloscope.	Observe waveform output on the oscilloscope, ensure that the oscillator is producing an 8MHz signal.
EC-06	Measure clock frequencies from ProMini using an oscilloscope.	Observe waveform output on the oscilloscope, ensure that the oscillator is producing an 8MHz signal.
EC-07	Test soldered pins on DWM1000 chips to ensure no short circuits.	Apply low voltage across two pins on the PCB. If there is no measured voltage and maximum allowed current across the pins, they are shorted and need to be resoldered. If the voltage is measured and stable, then the pins are safe.

Table 7: Electrical Test Cases

5 Software Test Plan

The software components consist primarily of a rust server running on a laptop, desktop, ipad or other single board computers. The server communicates with beacons via User Datagram Protocol (UDP). The users can view data and perform maintenance operations on the server through a simple web interface via common web browser such as Google Chrome or Firefox. Tests on the software side are primarily UI focused, which best emulate the user experience and involve all aspects of the system allowing a smaller set of tests to validate the entire system. Some additional automated unit tests are written primarily to test SQL queries for correctness, as they frequently break when the database schema is changed.

ID	Test Proceedure	Validation
SW-01	Build automated tests using Rusts built in testing framework to verify all Postgres queries are valid.	Execute the tests, verify that all test cases pass.
SW-02	Logging in as responder correctly uses the anonymous credentials.	Login in as responder, and attempt to create new objects or update existing ones, these operations must fail.
SW-03	Logging in as admin correctly uses the administrator credentials.	Login in as administrator, and attempt to create new objects or update existing ones, these operations must succeed.
SW-04	Beacons must be capable of sending location data to the server.	Login as administrator, select the diagnostics page, press the start emergency button and verify that data points start showing up on the page.
SW-05	The DPU must automatically connect with all beacons in the network.	Login as a responder, and automatically be directed to the map view page. Verify that clicking on a valid map name displays the beacons on that floor on the User Interface. Tags must not display on the map until an emergency is initiated.
SW-06	The server must show up to date position information about users.	Login as a responder, and navigate to the status page. In an emergency, the status page must show the ”Last Seen” column indicating the freshness of the information displayed.
SW-07	Administrators must be capable of creating and updating users, beacons, and maps as they expand their company.	Login in an administrator, and create a user, a map and a beacon. Then ensure the objects have been created in their respective list pages.

Table 8: Software Test Cases - Part 1

SW-08	The server must handle multiple beacon connections sending data for multiple tags at once.	Setup the system with at least 3 beacons and at least 2 tags, then view the map view or status pages and verify that the 2 beacons are correctly located.
SW-09	Maps on the User Interface must correctly use the pixel to distance scale to accurately display the tags and beacons on the maps.	Create a custom map as an administrator, upload the blueprint image and determine the scale in pixels per meter. Add becons to the map and then verify on the "Map View" page that the beacons and tags are in the right location.
SW-10	Adminstators must be capable of updating beacon and user information.	Login as adminstator and create an user and beacon. Edit user and beacon information. The user and beacon information should reflect the proper changes.
SW-11	Viewing the website should redirect users to see the login page immediately.	Enter the URL for the website. The option to select between admin login and responder login must appear.
SW-12	Website layout must scale properly on all types of devices. UI must be uniform without misplaced icons.	Enter website onto multiple devices browser. Website UI looks uniform and consistent among all devices.
SW-13	Improper admin login credentials must be rejected.	Login using incorrect login credentials. Error message must appear and client cannot login as admin despite multiple tries.

Table 9: Software Test Cases - Part 2

6 Usability Test Plan

The Akriveia Beacon's web server user interface will undergo usability testing to determine the state of its usability at various stages of development. The usability testing outlines testing procedure that will be done by the engineering and design team using heuristic usability evaluations. Each evaluator will independently examine the UI and check for compliance and usability. After collecting the results the team will discuss and compile possible solutions to usability issues and generate a list of solutions. Finally, the redesign will be implemented and regression testing will be done. Analytical usability testing will take place during the prototype and final product phases of development as the user interface is well defined during these two stages. The testing procedure will follow the steps described below.

Step 1: Usability Research Data Collection

The first step is to collect data generated by the usability test. Each evaluator will perform tasks outlined in the analytical usability testing procedure. From the procedures performed issues will be highlighted and documented. Each issue will have the following:

- An issue identification (ID).
- Note where it happened (screen, module, UI widget, flow, etc.).
- Task the user was engaging in.
- Concise description of the issue.

Data collected will be shown in a table similar to the table below:

ID	Where	Task	Description	P1	P2	P3
1	Login Page	Login with wrong Password	No error message for wrong user name input	X	-	-
2	Map View	Click on beacon icon	Beacon info text too small	-	X	X

Table 10: Usability Test Results

Step 2: Issue prioritization

Once sufficient testing has been performed by evaluators of the team, issues must be prioritized as time and resources are limited for this project. Each usability issue receives a grade of severity, influenced by factors such as:

- Task criticality: Impact on user if the task is not accomplished.
- Issue frequency: How many times an issue has occurred with various participants.
- Issue impact: How much has it impacted the user trying to accomplish the task.

Step 3: Solution Generation

After usability testing is performed following test cases described in section 5, using the combined feedback and evaluations, the engineers at TRIWAVE SYSTEMS will re-evaluate possible UI designs for each usability issue that occurred during testing to determine the best and optional solution. A list of recommendations and solutions will be generated with usability test results. For each design decision several alternative solutions must be generated to include other possible ways to address the issue.

Once internal usability testing is complete external usability testing will be carried out in cycles with real users consists of volunteer participants ideally relating to targeted audience. The first test cycle occurs near the end of the prototype phase and the second test cycle occurs near the end of the Final Product phase. Testing will be done with two small groups of participants that are unfamiliar with the project development environment. First group will be asked to perform usability test cases outlined in section 5. An observer will document actions and observations of the testing process as well as to keep note of average time to complete each task, the amount of errors and error rate, number of tasks completed, and perform a sequence analysis. Issues will be represented similar to the method mentioned previously. With the collected data the designers will re-evaluate the user interface and develop further design solutions for potential issues. After re-design and implementation of a second small group of participants will be asked to perform the same tasks as the first group to finalize design and changes.

From the results generated by participants the following usability elements will be addressed throughout the two testing cycles and development stages.

- **Easability:** The familiarity and intuitiveness of the system and how comfortable the users are with the user interfaces in general.
- **Navigation:** The reliability of the navigation sequences are, how easy is it for the users to understand paths, and/or short cuts. Can the users easily retrace their steps or go back to previous states if they have made a mistake?
- **Responsiveness:** Does the users receive sufficient feedback from interacting with the system?
- **Intuitiveness:** How quickly can a new user familiarize themselves with the user interface? Whether or not the users are able to perform tasks within a certain amount of time?
- **Robustness:** Safety and reliability of the device and system are addressed by eliminating or minimizing potential error (slips and mistakes) and enabling error recovery.

By following these usability testing procedures mentioned above, the engineers and designers at TRIWAVES SYSTEMS can ensure a reliable and intuitive user interface will be produced to meet the needs of its end users.

7 Conclusion

As urban centers around the world experience rapid growth and changes, so does the risk of being potentially trapped within buildings during disasters. The time period right after a disaster strikes is the most critical for saving victims. As such, a reliable and accurate indoor location rescue system is needed to aid first responders in locating trapped personnel. The Akriveia Beacon is a system of anchor beacons and ID tags designed to provide safe and reliable location information to first and rescue operations during such situations.

As a system to be used during emergency disaster situations, it is critical that each of the component and the system as a whole must be designed in such a way where potential risks and failures are minimized or eliminated. To validate designs and to identify potential issues, a comprehensive test plan was made to ensure all major system components are functional and adhere to design requirements, standards and regulations.

A detailed test plan was clearly outlined in this document to present the steps and processes in which the team will use as a reference to validate the Akriveia Beacon system. All major hardware, software, and electrical component of the Akriveia Beacon will be tested individually as well as integrated together as a system. Furthermore, to guarantee ease of use for the end users, a usability test plan is also provided. This test plan will ensure that the final Akriveia Beacon product will provide quality and assurance for its users.