**Portland State University**

**Electrical and Computer Engineering Department**

11

**Capstone 2011**

**TIU Tracking Project – Testing Documentation**

**Sponsor: Intel**

**Advisor: Prof. Robert Daasch**

**Members: Dung Le**

**Tri Truong**

**Lynh Pham**

**Man Hoang**

**Daniel Ferguson**

## Version History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Primary Author | Revision Date | Approved By | Approval Date | Description of version |
| 1.0 | Tri Truong | 4/18/2011 |  |  | Outline and test case descriptions |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table of Contents

[Version History 2](#_Toc290927367)

[1 Introduction 4](#_Toc290927368)

[2 Reference Documents 4](#_Toc290927369)

[2.1 Requirements Specification 4](#_Toc290927370)

[2.1.1 Functional Requirements 4](#_Toc290927371)

[2.1.2 Non-Functional Requirements 4](#_Toc290927372)

[2.2 Datasheets 5](#_Toc290927373)

[2.2.1 Microcontroller ATMega328P 5](#_Toc290927374)

[2.2.2 Wireless Transceiver RFM12B-S2 5](#_Toc290927375)

[2.2.3 Wi-Fi Transceiver Module WiFly 5](#_Toc290927376)

[2.3 Industry Standards 5](#_Toc290927377)

[3 Test Plan 5](#_Toc290927378)

[4 Test Cases 5](#_Toc290927379)

[4.1 Unit Test 5](#_Toc290927380)

[4.1.1 Hardware 5](#_Toc290927381)

[4.1.2 Firmware 6](#_Toc290927382)

[4.1.3 Software 6](#_Toc290927383)

[4.2 Integration Test 7](#_Toc290927384)

[4.2.1 Software Integration 7](#_Toc290927385)

[4.2.2 Hardware Integration 8](#_Toc290927386)

[4.2.3 Hardware-Software Integration 8](#_Toc290927387)

[4.3 Acceptance Test 8](#_Toc290927388)

[4.4 Parametric Test 9](#_Toc290927389)

[4.4.1 Device Dimension 9](#_Toc290927390)

[4.4.2 Accuracy of Locating Results 10](#_Toc290927391)

[4.4.3 Precision of Location Results 10](#_Toc290927392)

[4.4.4 Response time of Web-based Application 10](#_Toc290927393)

[4.4.5 Power Consumption 10](#_Toc290927394)

[4.4.6 Battery Life 11](#_Toc290927395)

[4.4.7 Detectable Range 11](#_Toc290927396)

[4.5 Functional Test 11](#_Toc290927397)

[5 Appendices 11](#_Toc290927398)

# Introduction

# Reference Documents

## Requirements Specification

### Functional Requirements

The key functionalities of the system are as follows:

* The primary user interface will be a web application
* Locations of TIUs will be displayed on an interactive 2D map within the user interface
* The user interface will show when an asset tag has low battery or when it is out of bounds

### Non-Functional Requirements

* The system will be able to track any TIU within a 10,000 sq ft area
* A tester in Intel lab occupies roughly 25 sq feet and the system will be able to locate the TIUs with an accuracy of 5 feet
* The locations of tracked TIUs will be updated within 10 minute intervals
* The asset tags must fit within a 1” x 1” x 1” dimension
* The asset tags will be permanently attached to the TIUs during operation
* The asset tags must use their own source of power. If the power source is a battery, then life must exceed one month
* The system will provide a user interface for adding and removing devices
* The system will provide a user interface to configure the geometry of the tracking area
* The locations in the database will be retained for 2 years
* The system will be able to operate properly in temperature range from -25°C to 85°C
* Parts and components will be available for manufacturing and maintaining
* All documentations regarding design, setup, use, and maintenance will be provided
* The system will not infringe upon existing patents, copyrights and trademarks and will comply with FCC rules and regulations

## Datasheets

### Microcontroller ATMega328P

<http://www.atmel.com/dyn/resources/prod_documents/8271S.pdf>

### Wireless Transceiver RFM12B-S2

<http://www.sparkfun.com/datasheets/Wireless/General/RFM12B.pdf>

### Wi-Fi Transceiver Module WiFly

<http://www.sparkfun.com/datasheets/Wireless/WiFi/rn-131-ds.pdf>

## Industry Standards

# Test Plan

# Test Cases

Test case properties: accurate, economical, limited in complexity, repeatable, appropriate, traceable, self-cleaning

## Unit Test

Purpose: A unit test is a complete test of a module’s functionality. In order to be a complete check, a unit test consists of a set of test cases each of which establishes that the module performs a single unit of functionality to some specification.

### Hardware

#### Power Supply Unit

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Lynh Pham | | | |
| Test Case Name | Power Supply Unit | Test ID #: | Pwr\_01 |
| Description: | Measure output voltage of voltage regulator. Input voltage is supplied by power supply in capstone lab. Record output voltage versus input voltage values. | Type: | white box  black box |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tester Information | | | | | | | |
| Name of Tester: | | Lynh Pham | | Date: | | |  |
| Hardware Version: | | **Voltage regulator Mic5205** | | Time: | | |  |
| Setup: | | Connect voltage meter to jumper VCC output (JP\_PWR) to measure output voltage | | | | | |
| Test | Input Voltage | Expected Output | Pass | | Fail | N/A | Comments |
| 1 | Check the red LED | Should be on with the connected jumper |  | |  |  |  |
| 2 | Connect 6V DC (4xAA batteries) to power jack | Output should be 3.3V+3% |  | |  |  |  |
| 3 | Connect 9V DC battery to power jack | Output should be 3.3V+3% |  | |  |  |  |
| 4 | Connect 9V adapter to power jack | Output should be 3.3V+3% |  | |  |  |  |
| Overall test result: | | |  | |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tester Information | | | | | | | |
| Name of Tester: | | Lynh Pham | | Date: | | |  |
| Hardware Version: | | **Voltage regulator XC6221** | | Time: | | |  |
| Setup: | | Connect voltage meter to jumper VCC output (JP\_PWR) to measure output voltage | | | | | |
| Test | Input Voltage | Expected Output | Pass | | Fail | N/A | Comments |
| 1 | Check the red LED | Should be on with the connected jumper |  | |  |  |  |
| 2 | Connect 6V DC (4xAA batteries) to power jack | Output should be 3.3V+3% |  | |  |  |  |
| 3 | Connect 6V DC battery to power jack | Output should be 3.3V+3% |  | |  |  |  |
| Overall test result: | | |  | |  |  |  |

#### Asset Tag

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Lynh Pham | | | |
| Test Case Name | Asset Tag Electrical Testing | Test ID #: | Tag\_01 |
| Description: | Checks the asset tag responding RSSI outputs. | Type: | white box  black box |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tester Information | | | | | | | |
| Name of Tester: | | Lynh Pham | | Date: | | |  |
| Hardware Version: | | Asset tag | | Time: | | |  |
| Setup: | | Remove the breakout board then connect the programmer cable to the SPI on the board. Power the testing board. Connect the base station to a PC by USB cable. | | | | | |
| Step | Action | Expected result | Pass | | Fail | N/A | Comments |
| 1 | Compile the final firmware version. | Arduino software should generate no warning/error |  | |  |  |  |
| 2 | Download firmware to MCU | AVR Studio should generate no error |  | |  |  |  |
| 3 | Turn off power |  |  | |  |  |  |
| 4 | Remove the cable from SPI |  |  | |  |  |  |
| 5 | Insert the breakout board into the tag |  |  | |  |  |  |
| 6 | Turn on power |  |  | |  |  |  |
| 7 | Open software controller | A new window should open |  | |  |  |  |
| 8 | Click Start | RSSI signal should be showed up on the window with Tag ID |  | |  |  |  |
| 9 | Repeat step 1-8 for other tags |  |  | |  |  |  |
| Overall test result: | | |  | |  |  |  |

#### Detector

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Lynh Pham | | | |
| Test Case Name | Detector Electrical Testing | Test ID #: | Dec\_01 |
| Description: | Checks the detector receiving RSSI from the asset tag. | Type: | white box  black box |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tester Information | | | | | | | |
| Name of Tester: | | Lynh Pham | | Date: | | |  |
| Hardware Version: | | Detector | | Time: | | |  |
| Setup: | | Connect the programmer cable to the SPI on the board. Power the testing board and an asset tag. Connect the base station to a PC by USB cable. | | | | | |
| Step | Action | Expected result | Pass | | Fail | N/A | Comments |
| 1 | Check the red LED | Should be on when turn the power switch on |  | |  |  |  |
| 2 | Compile the final firmware version. | Arduino software should generate no warning/error |  | |  |  |  |
| 3 | Download firmware to MCU | AVR Studio should generate no error |  | |  |  |  |
| 4 | Turn off power | The red LED should be off |  | |  |  |  |
| 5 | Remove the cable from SPI |  |  | |  |  |  |
| 6 | Turn on power | The red LED should be on |  | |  |  |  |
| 7 | Open software controller | A new window should open |  | |  |  |  |
| 8 | Check the green LED | Should be blinking with the connected jumper |  | |  |  |  |
| 9 | Click Start | RSSI signal should be showed up on the window with detector ID and Tag ID |  | |  |  |  |
| 10 | Repeat step 1-9 for other detectors |  |  | |  |  |  |
| Overall test result: | | |  | |  |  |  |

#### Proxy

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Proxy Electrical Testing | Test ID #: |  |
| Description: |  | Type: | white box  black box |

### Firmware

#### Asset Tag

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Dung Le | | | |
| Test Case Name | Asset Tag Firmware Test | Test ID #: |  |
| Description: | Test electrical connection on prototype board of asset tag. LED blinking program is flashed on chip to test if ATMega328P chip work properly. | Type: | white box  black box |

#### Detector

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Dung Le | | | |
| Test Case Name | Detector Firmware Test | Test ID #: |  |
| Description: | Test electrical connection on prototype board of detector. LED blinking program is flashed on chip to test if ATMega328P chip work properly. | Type: | white box  black box |

#### Proxy

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Proxy Firmware Test | Test ID #: |  |
| Description: |  | Type: | white box  black box |

### Software

#### Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Asset Tag Test #1 | Test ID #: |  |
| Description: | Test basic functions of controller software: support calibrating, location mode, transaction with database. | Type: | white box  black box |

#### Location Engines

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Tri Truong | | | |
| Test Case Name | Neural Net implementation Test | Test ID #: |  |
| Description: | Test the implementation of Location Engine using neural network concept. Faked data is fed to methods of the Location Engine. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Man Hoang | | | |
| Test Case Name | Propagation Loss Model Test | Test ID #: |  |
| Description: | Test the implementation of Location Engine using neural network concept. Faked data is fed to methods of the Location Engine. Check error handling when methods are called. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Dung Le | | | |
| Test Case Name | Euclidean Distance Method Test | Test ID #: |  |
| Description: | Test the implementation of Location Engine using neural network concept. Faked data is fed to methods of the Location Engine. Check error handling when methods are called. | Type: | white box  black box |

#### Web-based application

##### Server side

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Man Hoang | | | |
| Test Case Name | Server side | Test ID #: |  |
| Description: | Test basic functions of server side. Test response of the server to a request sent from client. Return data from server is faked (random). | Type: | white box  black box |

##### Client side

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Man Hoang | | | |
| Test Case Name | Web-based application – Client side test | Test ID #: |  |
| Description: | Test basic functions of client side of the Web-based application as User Interface | Type: | white box  black box |

## Integration Test

Purpose: Test of two or more modules or hardware/software together. Integration testing checks that the major modules of the overall system operate correctly together

### Software Integration

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Controller – Location Engine Integration | Test ID #: |  |
| Description: | Test the integration of location engines and the controller application by calling constructors, methods determined by the common interface. Fake data is created by the controller and fed to location engines. Check to returned result of location engines. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Man Hoang | | | |
| Test Case Name | Server-Client Integration test | Test ID #: |  |
| Description: | Test transaction between the server and client. Transacted data which is created by server is faked. The return location from server is used by client side to display the location of requested asset tag. | Type: | white box  black box |

### Hardware Integration

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Detector – Asset Tag Communication | Test ID #: |  |
| Description: | Test transmitting and receiving between a detector and an asset tag. The data packet with specified fields is checked. Detector is implemented on Arduino board. Serial Monitor is used to record the received data packet. Packet’s fields are check to ensure data integrity. | Type: | white box  black box |

### Hardware-Software Integration

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Calibrating Mode Test | Test ID #: |  |
| Description: | Test hardware-software integration in calibrating mode. Data flow is checked: data packet from asset tags through detectors to proxy, and then recorded by controller to database and calibrating data file. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Locating Mode Test | Test ID #: |  |
| Description: | Test hardware-software integration in calibrating mode. Data flow is checked: data packet from asset tags through detectors to proxy, and then recorded by controller. Locating data will be passed to location engine and location engine in turn return location of asset tags. Correctness of locations is not a concern in this test. | Type: | white box  black box |

## Acceptance Test

Purpose: An acceptance test is a formal document stipulating the conditions under which the customer will accept the system.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Man Hoang | | | |
| Test Case Name | Tracking System Test #1 | Test ID #: |  |
| Description: | Test the performance of the locating system: The system must return the current location of a requested asset tag with accuracy within 1.5m, including battery level of the asset tag. Information is displayed on a web-based application. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Lynh Pham | | | |
| Test Case Name | Battery Life of Asset Tag | Test ID #: |  |
| Description: | Measure the battery life of asset tags in normal operation mode. The battery of asset tags must last at least one month as requires. The stress test is used to evaluate battery life and conducted in shorter interval of time (2 days) than reality. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Tri Truong | | | |
| Test Case Name | System Deployment Time | Test ID #: |  |
| Description: | Measure deployment time of the complete system, including | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | User configuration – Controller Module Test | Test ID #: |  |
| Description: | Test usability of the step-by-step guide for the front-end software modules. Follow **Software User Guide** instructions to make sure the user guide is well-explained. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Tri Truong | | | |
| Test Case Name | Detector Installation Test | Test ID #: |  |
| Description: | Test deployed detectors which are mounted on wall/ceiling to ensure reliability, safety, and detectable range of detectors. Measure force required to detach detector out of wall/ceiling. | Type: | white box  black box |

## Parametric Test

Purpose: testing to determine or confirm detailed parameters of a completed design. The design must satisfy three figures of merit: accuracy, dimension of asset tags, and battery life of asset tag. Parameters of other modules should be also tested before delivering to our customers (Intel Sponsors).

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Tri Truong | | | |
| Test Case Name | RF Transceiver Detectable Range Test | Test ID #: |  |
| Description: | Test the detectable range of the locating system with various settings of broadcasting power of asset tags (from 0 dB to -20 dB). | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Tri Truong | | | |
| Test Case Name | Tracking System Cover Range Test | Test ID #: |  |
| Description: | Measure the range of tracking system in which an asset tag can be located. Conduct test to make sure any asset tag can be located in the setup area. | Type: | white box  black box |

### Device Dimension

#### Asset Tag

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Lynh Pham | | | |
| Test Case Name | Detector Dimension Measurement | Test ID #: |  |
| Description: | Measure size of all detectors. The size of detectors was not specified as a design goal. Dimension of detectors is reported as a part of deliverables. | Type: | white box  black box |

#### Detector

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Lynh Pham | | | |
| Test Case Name | Detector Physical Dimension | Test ID #: |  |
| Description: | Test the asset tags’ size to ensure it satisfies the requirements. The asset tags must fit into 1 x 1 x 1 cubic in. Measurement is conducted with 10 asset tags. | Type: | white box  black box |

#### Proxy

### Accuracy of Locating Results

### Precision of Location Results

### Response time of Web-based Application

### Power Consumption

#### Asset Tag

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Dung Le | | | |
| Test Case Name | Power consumption of Asset tag | Test ID #: |  |
| Description: | Measure current drawn by an asset tag in the operating mode (broadcasting signal), idle (stand by) mode, and sleep mode. | Type: | white box  black box |

#### Detector

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Dung Le | | | |
| Test Case Name | Power consumption of Asset tag | Test ID #: |  |
| Description: | Measure current drawn by an asset tag in the operating mode (broadcasting signal), idle (stand by) mode, and sleep mode. | Type: | white box  black box |

#### Proxy

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Daniel | | | |
| Test Case Name | Power consumption of Proxy Module | Test ID #: |  |
| Description: | Measure current drawn by an asset tag in the operating mode (broadcasting signal), idle (stand by) mode, and sleep mode. | Type: | white box  black box |

### Battery Life

#### Asset Tag

#### Detector

### Detectable Range

## Functional Test

Functional tests are conducted to demonstrate that the delivered tracking system satisfies requirements of the capstone project. Test cases are built based on requirements specification.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Man Hoang | | | |
| Test Case Name | Asset tag request Test | Test ID #: |  |
| Description: | Test response of the web-based application to random requests of users. The last known location of the request asset tags is displayed on 2D map. | Type: | white box  black box |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Writer: Tri Truong | | | |
| Test Case Name | Low Battery/Out of bound Alert Test | Test ID #: |  |
| Description: | Test response of the web-based application in the case a requested asset tag has low battery or cannot be detected due to out of range. Alert message is displayed on Web-app. The battery level is displayed to indicate batter’s status. | Type: | white box  black box |

# Appendices