# Test Plan Documentation

Test Plan Documentation

Smart House Project

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 23/09/15 | 0.1 | Structured outline. | Paul |
| 24/10/15 | 0.2 | Updating Information (updates are, from here on, denoted by green coloring) | Paul |
| 8/11/15 | 0.3 | Further explained the software testing specifications, and performance testing | Paul |
|  |  |  |  |

Test Plan

**1. INTRODUCTION**

We will be using iterative development like XP so our testing will be a concurrent process. Running along side our software development. Every software component will be tested upon completion and implemented immediately when testing is successful.

**2. OBJECTIVES**

The objective of our tests are to make sure every component is working properly and can communicate securely with a server.

We will be testing the hardware as well as the software components frequently.

**3. Required Resources**

Hardware – Arduino. Smart House, Multi-meter

Software – Processing, Arduino, Fritzing

Testing Tools – Fritzing, Processing, Arduino

Staffing – Paul, Jabir, Hassan

Testing Responsibilities:

P Hardware testing

J Software testing

H integration testing

**4. Test Case Development**

4.1 Hardware testing → Should we be testing the hardware?

Hardware testing will ensure us that our physical devices are correctly installed and ready to be used.

Hardware components are denoted by the letter H.

To be tested:

H1. Arduino

Arduino will be tested before every development session. Active sketch is to be re-uploaded every hour to ensure stability. A test sketch will be developed to initiate hardware signals and test the connected devices at a whole.

H2. Devices

Devices functionality will be established at the beginning of development session, but not during. Both Responsiveness and reliability of singular devices such as a LED, will be tested.

How the Hardware will be tested:

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality |  | How it will be tested | Priority |
| Responsiveness | Make sure every component works | With a Multimeter | Essential |
| Reliability | Using hardware component should yield consistent results | Test sketch created in arduino environment | Essential |

4.2 Software testing → unit testing

Software testing will be performs concurrently with software development to ensure that every iteration of a software component can be implement fully upon approved exit criteria.

Software components are denoted by the letter S.

Software to be tested:

|  |  |  |
| --- | --- | --- |
| Software to be tested | How they are to be tested | Priority |
| S1. Control Protocol | Arduino sketch to control and Devices | Essential |
| S2. Communication Protocol | Processing sketch used to Communicate with server | Essential |

S1. Control Protocol

Arduino sketch to control and interpret signals from Each individual device.

S2. Communication Protocol

A Processing sketch that will be used primarily to send and receive messages from the server.

How the software will be tested:

Performance testing

Performance Testing allows us to gain a better overview on how reliable and functional our software components are, and how well they execute their respective tasks or duties. Factors like stability, speed of communication, etc. are not immediately necessary for the components pass criteria. However, having a clear and concise understanding of our software components limitations become invaluable when the code inevitably breaks and needs to be fixed.

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality |  | How it will be tested | Priority |
| Stability | - | Compile the code | Essential |
| Speed | Find out how much time it takes to send/receive message | Group of functions | Non-Essential |
| Reliability | Consistent and reliable results from execution | Write a sketch to initiate a software component repeatedly | Essential |
| Safety | Secure communication | - | Essential |

4.3 Environment simulation

Our microcontroller will be interfacing with a server in the future so we will be using Processing sketches to simulate this interaction. This allows us to know that our information is being sent correctly.

Later on in the project, We will be receiving a mockup interface to test that our devices are sending and receiving Information correctly. We will be replacing our Processing sketch with this. Integration testing starts here.

**5. Pass/Fail Criteria**

A unit passes once all criteria is met and can be ready to implement.

5.1 Pass criteria – Hardware

Device provides basic functionality

All values read need to be consistent and reliable.

5.2 Pass Criteria – Software

Software component is successfully compiled and can be implemented.

Using software component should yield predictable results when run.