3/23/2017

Lecluse, Herm H.

mINOR i&r

Test Plan

Internet of Trash



Contents

[Introduction 2](#_Toc478075767)

[Objectives and Tasks 2](#_Toc478075768)

[Objectives 2](#_Toc478075769)

[Tasks 2](#_Toc478075770)

[Pre-Testing 2](#_Toc478075771)

[Testing 2](#_Toc478075772)

[Post-Testing 3](#_Toc478075773)

[Problem Reporting 3](#_Toc478075774)

[Scope 3](#_Toc478075775)

[Testing Strategy 3](#_Toc478075776)

[Unit Testing 3](#_Toc478075777)

[System / Integration Testing 3](#_Toc478075778)

[Performance Testing 4](#_Toc478075779)

[Hardware Requirements 4](#_Toc478075780)

[Control Procedures 4](#_Toc478075781)

[Approvals 4](#_Toc478075782)

# Introduction

This test plan contains information about how to test different sensors to calculate an accurate distance to a specific object. Three different sensors will be tested on:

* Accuracy based on how accurate the measurement is
* Efficiency based on how many times the sensor was accurate
* Functionality based on usability (think of processing time, size, etc.)

# Objectives and Tasks

This chapter will contain information about the objectives of this test plan are. Tasks in this test plan will be addressed as well.

## Objectives

The main objective of this document to create a test plan, to determine which sensor can detect the distance between the sensor and a distance best. This must be done in a sealed container at maximum one meter distance between sensor and object. Which after been collected should be transformed into a specific fill level of the specified container.

Another objective is to calibrate the different sensors so that a translation from raw value to usable information can be made.

A last objective is to see how to connect to the LoRaWan network and Transfer data over it.

## Tasks

The tasks that will have to be completed can be divided in four sub tasks, which are listed below.

### Pre-Testing

This task is the preparation of the test setup and a control test will be measured with an analogic tool (e.g. Tape Measure). The first test will be just as simple as a cardboard shoe box, which will have a raiseable bottom. This way the ‘fill level’ of the box can be virtualized. And will be replaced with another design which yet should be decided on.

### Testing

* Testing of the Laser sensor
* Testing of the Infra-red sensor
* Testing of the Ultra sonic sensor
* Posting crafted data to the LoRaWan network
* Testing data parsing algorithms.

### Post-Testing

In this task, the tests results will be reviewed with these from the control test. Evaluation takes place here as well. The result of the test will be determined here.

### Problem Reporting

This is a special task; this task will be executed throughout the whole test. Every problem that occurs should be documented in a document which contains the following properties;

* Date of occurrence
* Description of what happened
* Name; who found the problem
* Solution (not mandatory)
* Description of how to recreate problem.

# Scope

The scope will determine what will and what will not be tested in this test plan.

The test plan will not include any other sensors than listed in the introduction of this document. Furthermore, is this a test plan for scaled down versions of sensors that could be used in the real environment. Since the sensors are scaled down in terms of reach and size just like the ‘container’, this should not be a problem. At last any other unforeseen situation is per definition out of scope, unless discussed with the group.

# Testing Strategy

This section will explain how and what will be tested on a technical level. This means that the manner of unit testing, System/integration testing and performance testing will be explained here.

## Unit Testing

The sensors logic will be tested using unit tests. Hence that only the logic about parsing data will be tested, NOT the actual sensor itself.

Herm Lecluse will be responsible for the unit tests with cooperation of Rick van Osch and Loek Ehren.

The scripts regarding this parsing / normalization will be written by any of the people listed above. Unit testing will be done on every pull request to the ‘IoTHardware’ project. This way we can determine if a piece of software is tested and is working.

Code Coverage >85%!

## System / Integration Testing

The transmitted data should be in a list on a server, this can be tested.

Rick van Osch will be responsible for checking IF at least ANY data has been received in the backend. Rick is also responsible for the link between the LoRaWan network connection to the backend.

By sending data over the LoRaWan network we can check if data has been received and so can be processed. So, the process is: Send data -> Receive -> Check if data is received. These tests will be done on a Daily base.

## Performance Testing

By instantiating multiple virtual sensors, we can pretend that there are “x” – number of sensors sending data to the system, this way we can see how well the server handles with many requests.

For the Test 50 sensors will be transmitting data simultaneously.

Loek Ehren will be responsible for writing a script that starts 50 tasks almost at the same time and send data to the server. This be tested during working hours, that way the system can be tested on maximum number of clients during peak time of the day.

# Hardware Requirements

To execute these test, the following equipment is needed:

* Arduino Uno / The Things Uno
* Raspberry Pi 3
* Infrared Sensor
* Laser Sensor
* Ultra-sonic Sensor

# Control Procedures

The results of these tests should be stored and monitored.

Mike Schatorjé (Project Manager) will keep an eye out for this reporting. He is responsible for this activity, as in this deliverable will show what kind of problems occurred during the testing. And so is one of the most important artefacts of the tests.

Herm Lecluse (Product Owner) oversees the Change Management cycle, and so, is in charge of a Change report if any were to be made.

# Approvals

Name (In Capital Letters) Signature Date

1. MIKE SCHATORJE

2. SIMONE FRANCESCONI

3. LOEK EHREN

4. RICK VAN OSCH

5. HERM LECLUSE