Test Plan

Editor: Pontus Erlesand

Version 0.1

Status

Reviewed		
Approved		

TSKS05
Pontus Erlesand

TSKS05-POZYX
pozyx.cdio@gmail.com

LIPs
Page 1

PROJECT IDENTITY

2016/HT, TSKS05-POZYX
Linköping University, (ISY)

Group members

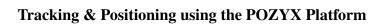| Name	Responsibility	Phone	Email
Rasmus Vilhelmsson(RV)	Project Leader (PL)	0700 423 400	rasvi146@student.liu.se
Pontus Erlesand(PE)	Head of documentation (DOC)	0707 964 955	poner538@student.liu.se
Markus Petersson(MP)	Head of sensor fusion	0703 823 630	marpe163@student.liu.se
Ching-Hsiang Yang(CY)	Head of hardware	0737 227 489	chiya469@student.liu.se
Susmita Saha(SS)	Head of testing	0720 451 029	sussa301@student.liu.se

Group e-mail: pozyx.cdio@gmail.com
Customer: ISY, Linköping University, 581 83 Linköping
Customer contact: Danyo Danev, 013-281335, danyo.danev@liu.se
Course leader: Danyo Danev, 013-281335, danyo.danev@liu.se
Supervisor: Trinh Van Chien, 013-282643, trinh.van.chien@liu.se

Contents

Document history

Version	Date	Changes	Sign	Reviewed
0.1	2016-10-31	First draft		PE

1   Introduction

In this document the project group will present how the testing will be carried out, to confirm certain milestones and requirements. The testing schemes to confirm the milestones will be carefully specified in this document. Testing that will be carried out to confirm each item in the Requirement Specification will be of a simpler manner within the group.

2   Testing methodology, milestones

The whole testing procedure is divided into individually testable items. Each test corresponds to one or several milestones defined in the project plan, and are designed to verify that the system adheres to the requirements described in the specifications. We present the proposed tests for each milestone in the sections 2.1 through 2.6. In table 1 the milestones intended for testing is presented, note that the last milestone does not require testing.

No.	Proposed Test Summary	Corresponding Milestone
1	Send known data	Receive sensor data on the computer
2	Receive data in MATLAB	Receive all sensor data and input them into MATLAB
3	Generate dummy data to test algorithms	Positioning and tracking with dummy data
4	Display estimated data on GUI	Display estimated path and position on GUI
5	Test with predefined paths	Pos. and tracking using real data in real time (GUI)
6	Complete system integration	Complete system integration
7	None	Complete all documentation

Table 1: List of milestones and proposed tests for verifying them

2.1   Send known data

We test the link between our device and the computer by sending known data from the device. By this we can simply conduct a comparison between sent and received data in order to verify that this milestone is fulfilled.

2.2   Receive data in MATLAB

We intend to test this by sending known test data. That way we can simply conduct a comparison between sent and received data (this time presented in MATLAB) in order to verify that this milestone is fulfilled.

2.3 Generate dummy data to test algorithms

For this test, we generate some artificial data, such as a constant turn corrupted by White Gaussian Noise. In this case we know what the estimated trajectories and positions should be for every time instant since we are the ones that are creating the "real" data. There are also some available examples from the tutorial sessions in courses such as *Digital Signal Processing* and *Sensor Fusion* that can be used for verifying the implemented algorithms. A constant turn trajectory could for example be generated using the code below.

```
traj=[10+5*cos(0:0.01:1.5);5*sin(0:0.01:1.5)];
```

2.4 Display estimated data on GUI

To test this we use dummy data to make sure that the tag can move smoothly over the map on the GUI. With this test in place we can easily see how well the processed data estimates the path of the tag and its position.

2.5 Test with predefined paths

To test the device in real time, we setup paths in the test environment and walk with the tag. We can compare the path displayed on the GUI with the paths designed. If the tag moves smoothly on the GUI while we move it around in the corridor, the processing is good enough. In this test we will also add the sensor data to the GUI and make sure it is presented well.

2.6 Complete system integration

In this test we will make sure that the modules of the system works as intended, and that the system can perform as expected. We setup the system and record a video of a person holding the tag while walking around the test environment randomly. We verify the estimated path by comparing it with the trajectory of the walk, as recorded in the video.

3 Testing methodology, requirements

Simple testing will be carried out within the project group to verify all requirements. At least two people will be present to test/verify each item. If any requirement testing is of interest for the customer or supervisor the group will show the test in person or present the result.

If the group notices that one, or several, requirements is of high importance the group will then make a more elaborate test and report the results.