Design Sketch

Editor: Pontus Erlesand

Version 1.0

Status

Reviewed	Pontus Erlesand	2016-10-04
Approved	Danyo Danev	2016-10-07

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	2016-09-28	First draft		PE
0.2	2016-10-04	Revised version		PE

1 Introduction

This document is submitted as part of the course TSKS05 and is indended to give an overview of the solution proposed to solve the given task. In section 3 we give a modular overview of the proposed system design. In sections 4 and 5 we provide additional details about the hardware and software modules respectively.

2 Definitions and terms

Term	Description
2D	Two dimensions
3D	Three dimensions
LOS	Line of sight
NLOS	No line of sight
KF	Kalman filter
EKF	Extended Kalman filter
UKF	Unscented Kalman filter
PF	Particle filter
IMU	Inertial measurement unit
GUI	Graphical User Interface
UWB	Ultra Wide Band
USB	Universal Serial Bus

3 Overall design

The proposed system is divided into modules with different purposes. On the top level we have two modules, the hardware module and the software module. The hardware module handles all things related to the arduino and the Pozyx platform. The software module is divided into two sub modules, the GUI module and the Data processing module, whereas the Data processing module is divided into two smaller sub modules as well. An overview of the proposed modular design of the system can be found in figure 1

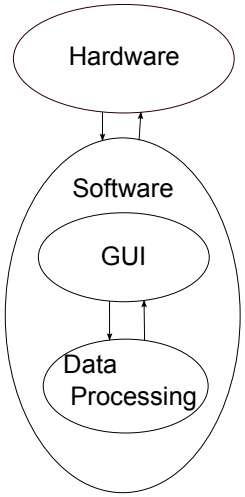

Figure 1: Overview of the system.

An overview of the information flow in the proposed system can be found in figure 2.

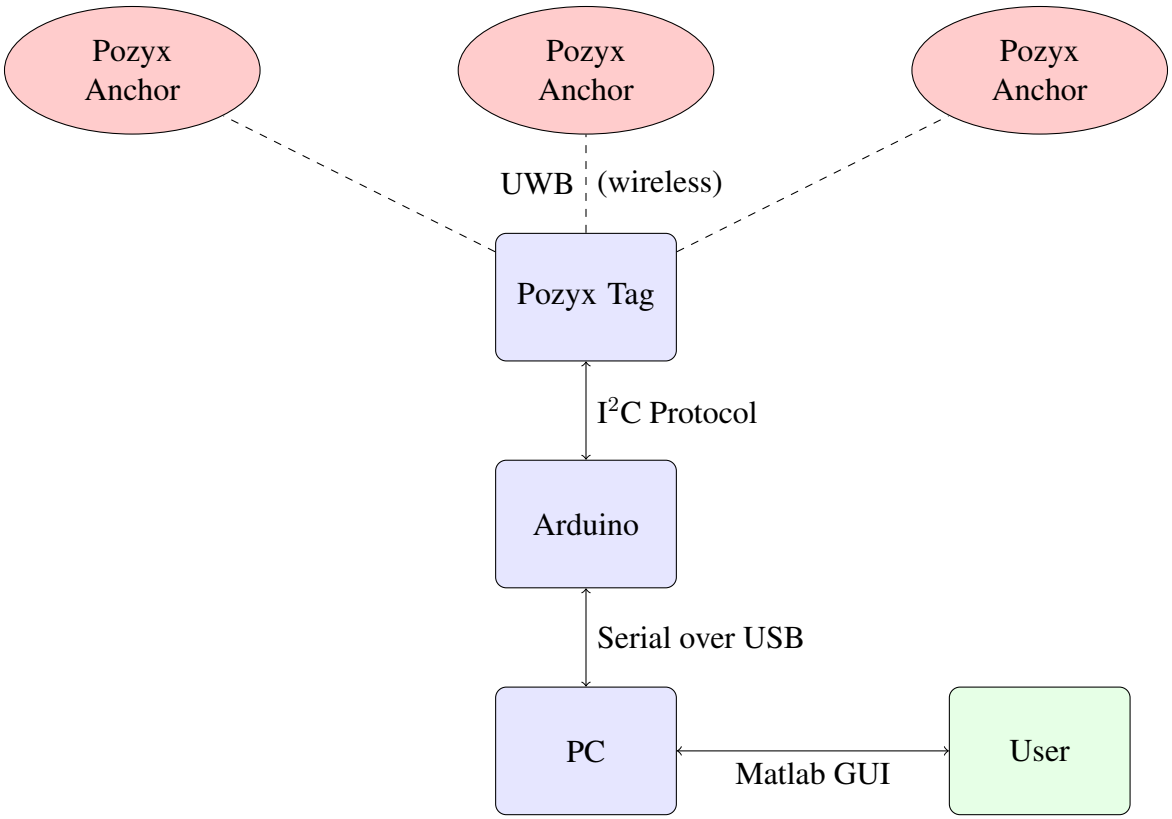

Figure 2: An overview of the information flow in the proposed system

3.1 Overview of module structure

In this section we give an overview of the proposed system design. We present the proposed modular structure as well as an overview of how the communication will work and in what environment the system will be deployed.

3.2 Communication

The communication between the anchors and the Pozyx tag is through UWB, from the tag to the Arduino the data is transferred through I^2C and finally from the Arduino to the PC the data is transferred through USB. This can be seen in figure 2.

3.3 Workspace frame

The workspace frame will consist of the Communications division's corridor, six anchors will be carefully positioned throughout the corridor. A map of the Communication Systems division's corridor is shown in figure 3.

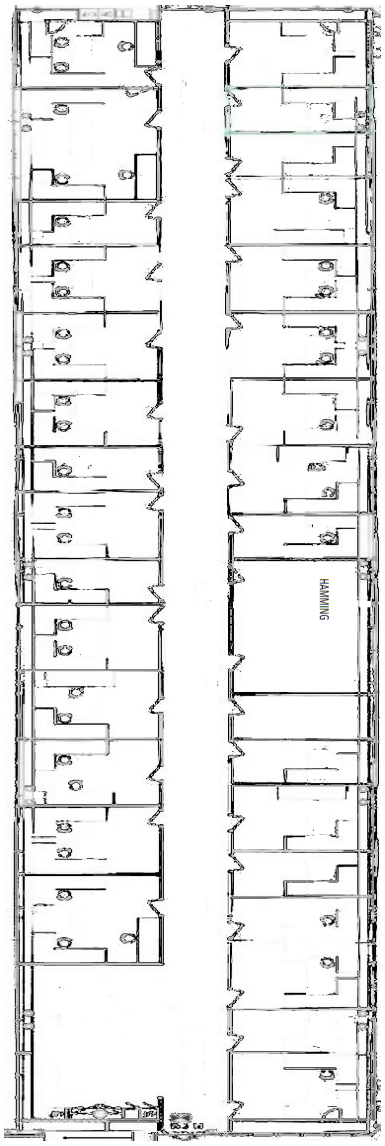

Figure 3: Map of the Communication Systems division's corridor

4 Hardware design

The hardware of our system consists of Pozyx anchors, Pozyx tags, Arduino boards and a computer.

4.1 Pozyx anchors

The anchors are powered by USB, but otherwise operate without wired connection. They are fixed in the room, preferably at positions with LOS to the tags for positioning and tracking. They communicate with the trackers and provide reference points for positioning and tracking.

The Pozyx specification demands at least three anchors for 2D tracking, and at least four for 3D tracking.

4.2 Pozyx tags

The tags are equipped with the same UWB modules as the anchors, as well as sensors such as accelerometers, gyroscopes and more. These sensor data are what we use to provide positioning and tracking functions.

The tags are designed to work with Arduino boards.

4.3 Arduino

Arduino boards are used to provide an interface between the Pozyx tag and the computer. The tags are soldered on the Arduino boards, and the I^2C interface is used to send data from the tag to the Arduino board.

The Arduino includes a chip to emulate serial port communication over USB, so we can connect the Arduino via USB to the computer for further data processing.

4.4 Computer

We use a commodity computer for the backend of our system, processing the data using Matlab.

5 Software design

The software module encapsulates two sub modules, the GUI and the Data processing sub modules. An overview of the entire software module can be found in figure 4.

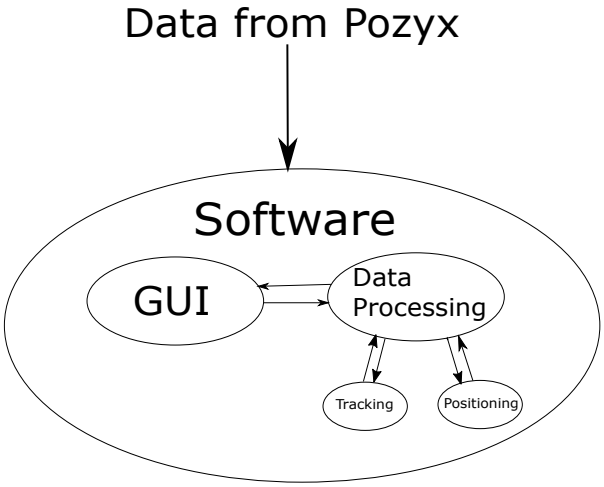

Figure 4: Overview of the software module.

Additional information about the GUI and Data processing sub modules can be found in sections 5.2 and 5.1 respectively.

5.1 Data Processing

The data processing sub module consists of two smaller sub modules called the positioning module, and the tracking module. Additional information about these two modules can be found in sections 5.1.1 and 5.1.2 respectively.

5.1.1 Positioning

The purpose of the positioning module is to estimate the positions of a tag.

5.1.2 Tracking

The tracking module can in some sense be seen as an extension of the positioning module. What it intends to do, that the positioning model doesn't is to incorporate a model for the movement of the tag into account, i.e. it provides rules and procedures for combining the estimated position of the tag, with the expected position that it should have according to the model.

The implementation of the tracking module is proposed to be object oriented. The tracking module should contain objects of the class *tracker* that should keep track of estimates and do tracking using different algorithms, e.g. KF or EKF. These objects are in turn fed data by the higher level tracking module. An overview of the tracking module can be found in figure 5.

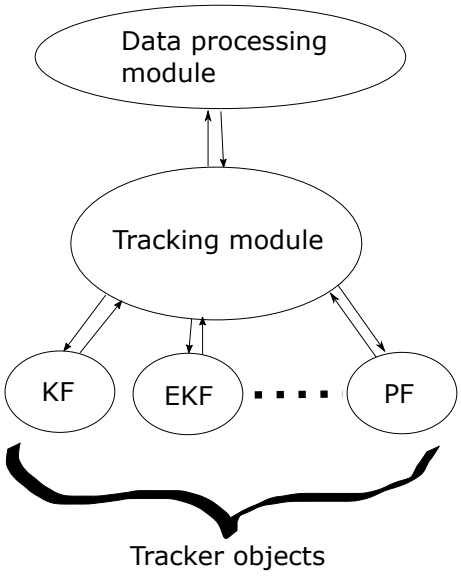

Figure 5: Overview tracking module.

5.2 GUI

The purpose of the GUI is to represent the estimated position of the tag on a computer screen for the user. This will be done using Matlab and the Matlab GUI toolbox with a object oriented programming implementation.

Following is a table of Classes that will be used for the GUI.

Table of classes for the GUI			
Class	Variables	Functions	Description
Circle_tag	x_pos,y_pos,radius	get_xy_pos	used to draw a circle on indicate where the tag is
Anchor	x_pos,y_pos	place_anchor	used to draw the position of the anchors on the map
Map	map_picture, Circles_vector , Anchors_vector	draw	draws the map, anchors and estimated position of the tag
Sensor	x_axis, y_axis, sensor_name	add_data	represents data from a given sensor

Following is a table of functions that is used for the GUI.

Table of functions for the GUI			
Function	Input	Output	Description
get_xy_pos	none	xpos , ypos	returns the estimated postition of the tag
get_sensor_data	none	sensor1,sensor2,sensor3	outputs data from all the sensors
set_xy_pos	xpos,ypos	none	gives the tag a new position on the GUI
add_data	sensor_value	none	adds new data to the sensor plotts
place_anchor	x_pos , y_pos	none	gives the anchor a position on the map

This is a preliminary idea of how the GUI part of the system will work and look like. Figure 6 roughly illustrates how the position of the tag will be represented on a map and how the sensor values will be represented.

Figure 6: Example of the GUI