The Indoor Orientation of Autonomous Mobile Devices for Increasing Human Perspective

Jiri Kotzian, Jaromir Konecny, Ondrej Krejcar, Tomas Lippa, Hynek Prokop, and Marian Kuruc

VSB-Technical University of Ostrava, Department of Measurement and Control, 17. listopadu 15, 70833 Ostrava-Poruba, Czech Republic
{jiri.kotzian,jaromir.konecny.st,ondrej.krejcar, tomas.lippa.st,hynek.prokop.st,marian.kuruc.st}@vsb.cz

Abstract. During recent years, rises in terrorist attacks and armed conflicts have increased the demand for autonomous devices. The need for devices with the ability to detect toxic gases, to be fire resistant and to multifunction has increased. We propose a concept of two such devices with the ability to comfortably and remotely control such devices and even with an autonomous control in remote areas inside the buildings. The localization by WiFi is used to locate a position where the GPS signal is not well presented. The ability to locate a mobile device by a wireless network is a well known possibility. Position information tools are currently used in many current areas. The main area of interest is in the use of locating and tracking users of a mobile information system to prebuffer large amounts of data to them before usage. All large data files are stored as artifacts along with its position information in a building or a larger area. The accessing of prebuffered data on mobile devices can highly improve response time needed to view large multimedia data. This fact can help with the design of new full scale applications for mobile devices.

Keywords: Orientation, Navigation, Embedded system, Wireless communication.

1 Introduction

The usage of mobile devices for orientation in open spaces has increased. There are several global navigation systems like GPS, Glonass etc. Navigation systems are very helpful in our everyday lives. The problem is rising in places with a high density of buildings. The precision of computing positions is too low. Inside buildings the navigation is usually not possible at all. The reason is low signal or total signal absence.

Different technologies for the navigation of mobile devices have to be used in buildings. For example: the human body uses stereovision for environment detection and orientation in cooperation with "maps" or other information (info panels, labels and indicators). Unfortunately this method is over the computation/power/space possibilities of today's embedded systems in mobile devices. It is also possible to equip rooms of a building with a set of transmitters like GPS. But this method is very expensive and complicated. The best method would use the current data infrastructure of the building – the net of mobile Ethernet access points (WiFi, WiMax,...). This

P. Chatzimisios et al. (Eds.): MobiLight 2010, LNICST 45, pp. 755–768, 2010.

method is described in the following paragraphs. A net of access points is not sufficient on its own. For obtaining the right position of mobile devices, it is necessary to equip the device with other sensors and maps. When needed we can dynamically place additional access points to achieve a higher communication range or a higher precision of position detection.

1.1 Need of Orientation and Navigation in Buildings

In our lives, situations can occur when humans need to see more and beyond. Some places are hidden to human vision, are too dangerous or too far away. There is a need to equip the user with some devices which increase human perspective. A special set of mobile devices was built at the Technical University of Ostrava for this purpose. These mobile devices allow the user to get more information about remote places. The first small handheld mobile device is the remote control and provides Human-Machine Interface to the user. The remote control is used for monitoring and controlling the second mobile device – a probe. The two devices are connected using wireless connection.

The probe is equipped with a group of distance, pressure, temperature and other sensors for environment detection. This device is able to move by itself to a desired position. This mobile device can search for people who have been trapped by an earthquake using infra camera and make audio-visual contact before the rescue come. Another example is searching for dangerous chemicals or to find criminals before authorities arrived; this is safer for the control staff. The simplest usage is to send or bring some item to a given position. Remote mobile robots are a common thing now, but this device should fulfill the given task using its own artificial intelligence at the end of the development. In the current phase the mobile robot is able to choose a way to its desired position using preprogrammed scenarios. The current problem is the right navigation in buildings. The set is displayed in (Fig. 1).

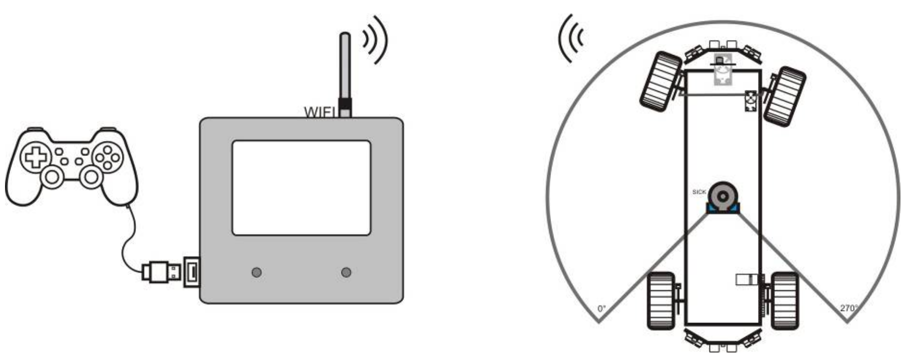

Fig. 1. The set of devices for increased human perspective. On the left side is the remote control with a gamepad in the case of manual mode. On the right side is the probe with sensors and drives. Communication is done by wireless communication.

2 Remote Control

The remote control is the first mobile device from the set. The set should increase human perspective. The remote control provides Human-Machine Interface. The remote control is a medium sized handheld device. It is displayed in (Fig. 2).

Fig. 2. Human-Machine Interface: Operation panel with color LCD display and touch screen. The controlled probe with sensors and actuator is in the back.

Remote control provides:

- monitor and control of the probe
- display information from sensors
- setting the task or the desired position in auto mode
- manual control using a gamepad or joystick
- maps insertion and actualization
- audio-visual interface
- diagnostic interface with trends and help

The user has full control of the second mobile device (the probe) or he can give the task to the probe. The user can handle the probe using a color display and touch panel. Manual mode is also available. The probe is controlled by a gamepad in manual mode. Programmed application provides intuitive user interface with a set of buttons, value displays and screens. Several screens are displayed in the following pictures. (Fig. 3 to Fig. 6).

First picture (Fig. 3.) displays the situation measured using distance sensors – laser, optical, and pieso. In the situation something is very close to the back of the probe. The only free space to ride is on the right side.

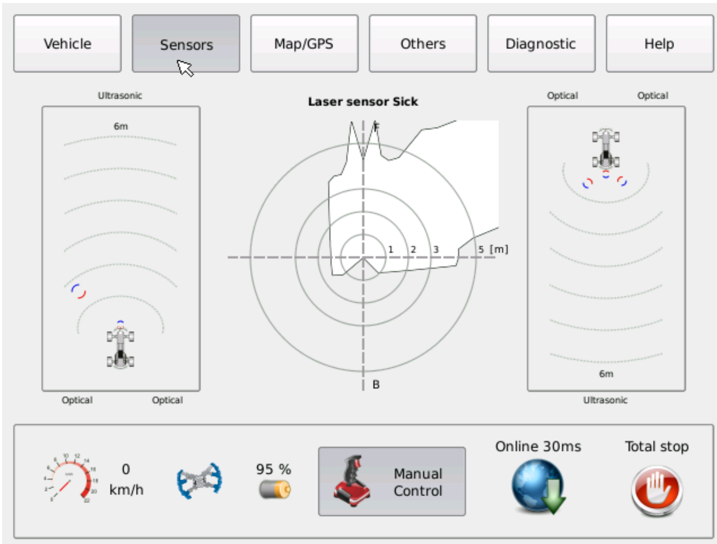

Fig. 3. Sensors screen from remote control application – laser, pieso and infra distance sensors

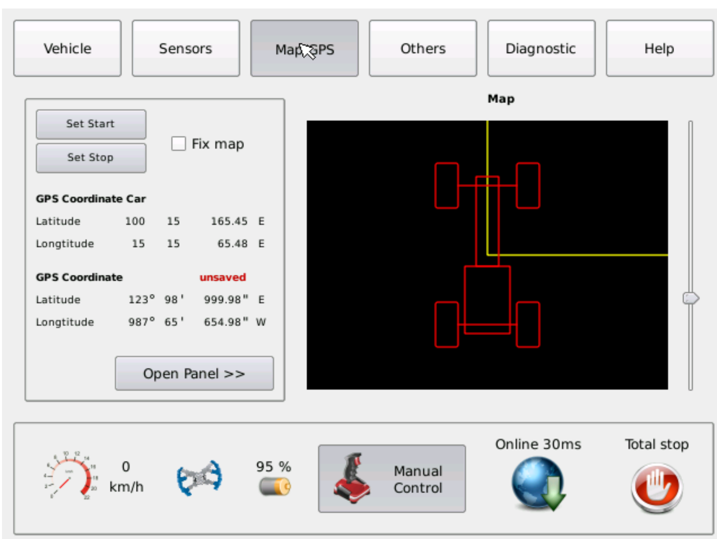

Fig. 4. Position of the probe and maps of remote control application

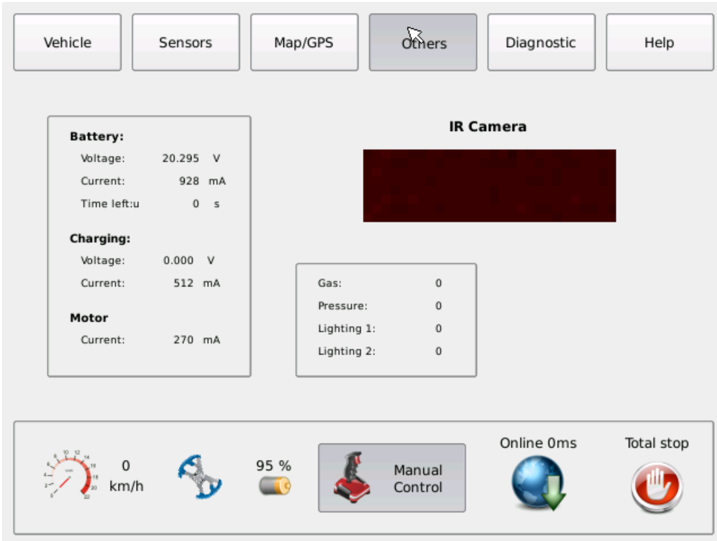

Fig. 5. Camera and diagnostic screen from remote control application

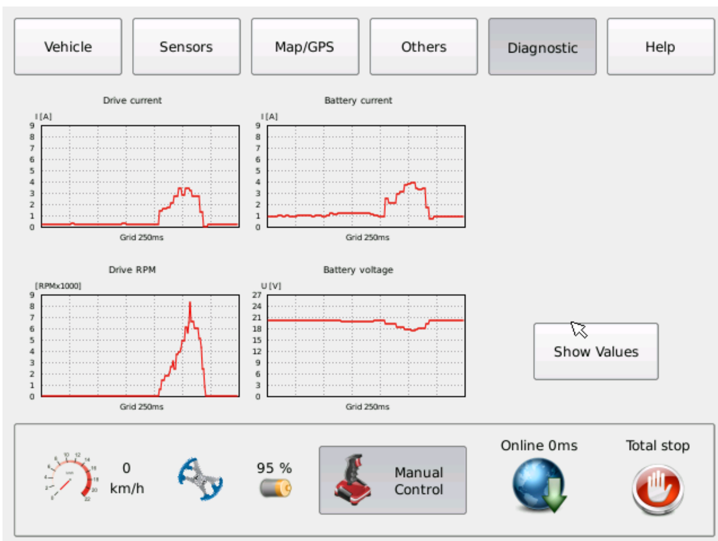

Fig. 6. Diagnostic screen from remote control application – trends or list of values

Next three pictures (Fig.4 to Fig.6.) illustrate other screens of the remote control.

2.1 Architecture of the Remote Control

The remote control uses the iMX31LiteKit embedded controlling board based on the ARM architecture [13]. The core of the board is the Freescale iMX31 microprocessor.

The embedded board is equipped with a set of interfaces (Ethernet, serial, SPI, SD, CF etc.) The controlling application is stored on external SD card. For debugging purposes, the device is equipped with Ethernet connection. The remote control communicates with the probe using the WiFi module Owspa311 connected via serial interface. The remote control uses a touch panel placed on the color display with a resolution of 640x480 pixels. It is possible to use a gamepad or joystick in the case of manual mode. The architecture is displayed in (Fig. 7).

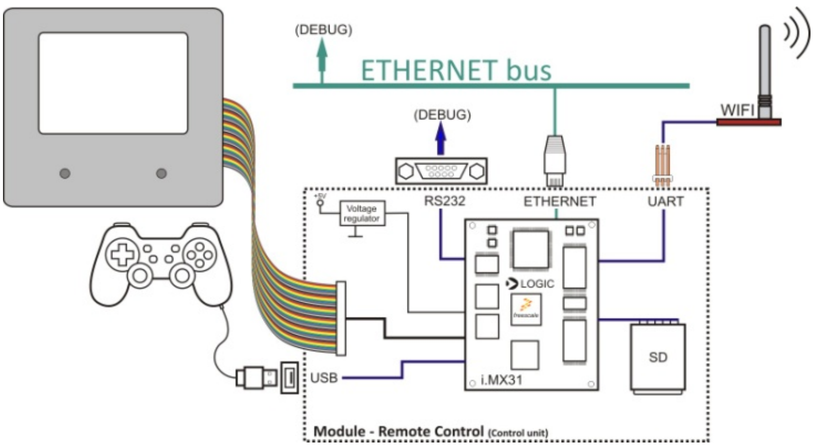

Fig. 7. Architecture of the Remote Control – display, mainboard and WiFi communication

2.2 SW Architecture of the Remote Control

The software application is based on the LinuxLink embedded linux [11]. The application is programmed and compiled using the TimeStorm integrated development environment (IDE) including board support package (BSP) for the iMX31Litekit [12]. The Fedora Linux host machine and the NFS (network file system) are used for developing the application. This way is quite complicated but very fast. The application is compiled and stored on the NFS or on an SD card. After reset (power on) the Logic Loader loads the application from the SD card and starts it.

The application is developed in the Qt graphic tool – Qt Creator. In this tool it is possible to create the design, windows, buttons and main root. To create the final application it is necessary to select an external compiler and to make the application. Another way is to use the Qt designer, which only generates functions and windows. The application is then programmed and compiled in some IDE environment – the TimeStorm in the case of the iMX31.

The QT is a general graphic library which can be used in several operating systems (Windows, X11, Linux). The library is simplified for embedded systems. All necessary servers are already included in the library. The application writes directly to the frame buffer (Fig. 8).

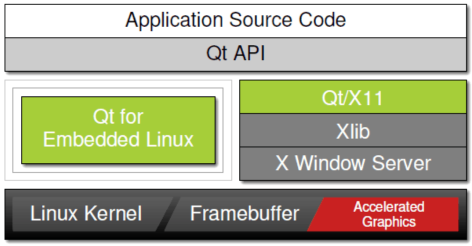

Fig. 8. SW Architecture of control application of the Remote Control

3 Probe

The second mobile device – the probe – is based on a massive chassis with a distributed control system, motor and high capacity battery. It is approximately 85cm meter long and 60cm high without an antenna (Fig. 9). The weight is approximately 10kg including the DC drive, the battery, the CAN based distributed control system and all sensors.

Fig. 9. Probe: Wheeled mobile device equipped with set of sensors

The probe includes a set of sensors for distance measuring, environment measurement and position and movement detection. For example, the Laser scanner can measure an environment up to 20meters in 270 degrees. The probe includes infra-camera, pieso and optical distance sensors, pressure, temperature, GAS sensors, 3-axis accelerometers etc. The audio-visual interface is in the preparation phase.

3.1 Architecture of the Probe

The probe is equipped with a distributed control system with a set of embedded control and monitoring boards. The system is based on the industrial CAN bus and the CANOpen application layer [4] [14].

The main control unit uses the same HW architecture like the remote control. The control unit is based on the iMX31LiteKit and it is also equipped with LinuxLink (Fig. 10).

Other control boards are based on industrial microcontrollers Freescale HCS12 with the cooperation of the FreeRTOS operating system. These boards are programmed using C programming language using the CodeWarrior IDE. The probe communicates with the remote control using the WiFi module Owspa311.

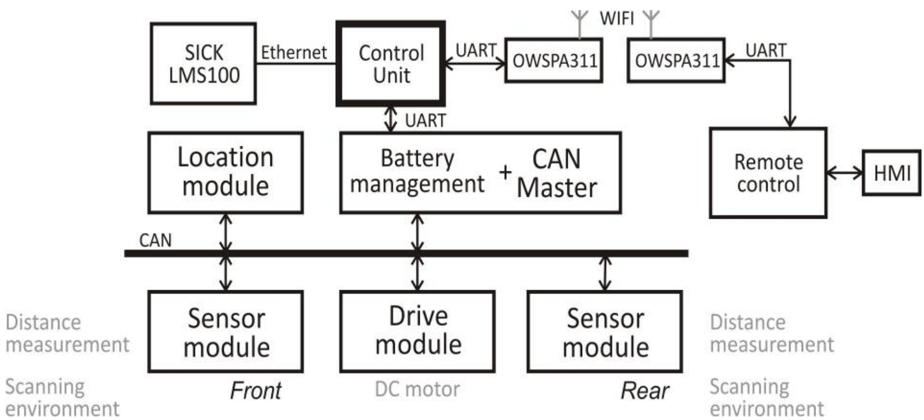

Fig. 10. Architecture of the probe: CAN based distributed control system

3.2 SW Architecture of Main Control Unit of the Probe

The main Control unit of the Probe uses a similar SW platform to the remote control. The software application is based on the LinuxLink embedded linux. The application is programmed and compiled using the TimeStorm IDE environment including BSP for the iMX31Litekit. The Fedora Linux host machine and NFS system is used for developing the application. The application is compiled and stored on the NFS or on the SD card. After starting up, the Logic Loader loads the application from the SD card and starts it (Fig. 11).

The main control application uses several cooperation processes. This method enables the main control system to dynamically start, stop or replace part of the application without influencing the rest of the application.

Used processes:

- vehicle_init – init of the application
- vehicle_guardian – observes run of processes – restarts if not responding
- vehicle_memory – data block – all values of the probe

- vehicle_slave – communicates with slave units and writes data to the shared memory
- vehicle_wifi – communicates with the remote control
- sick – reads data from sick laser scanner –writes data directly to the shared memory
- vehicle_control – main control, finds the path, maps.

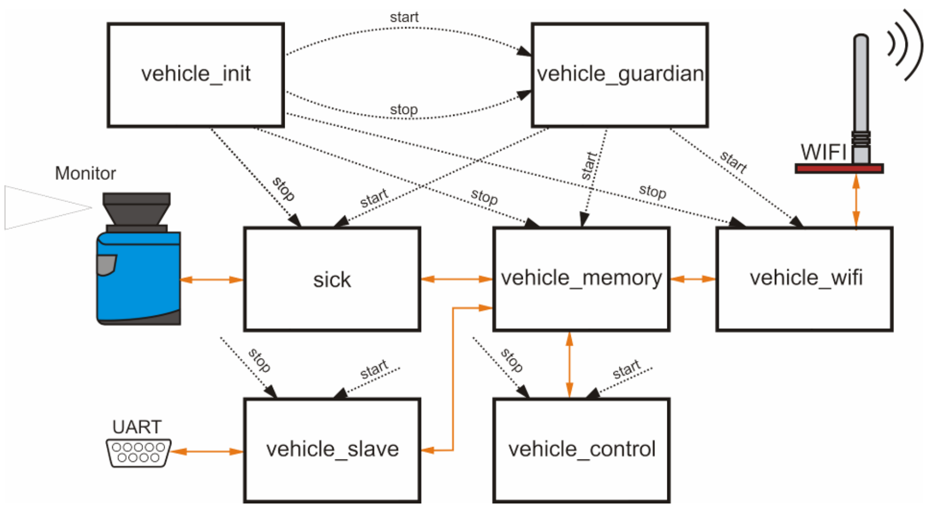

Fig. 11. Architecture of the probe: CAN based distributed control system

4 Localization

Another important part of the project is based on indoor localization. The primary localization is needed to detect a position inside the building. Consequently, the map of the detected location is loaded into a mobile device to activate other sensors to "open the eyes" of our mobile devices. The localization is made through a WiFi infrastructure.

If the mobile device knows the position of the stationary device (transmitter), it also knowsits own position within a range of this location provider. The typical range varies from 30 to 100 m where there is WiFi, respectively 50 m where there is BT case or 30 km for GSM. Granularity of the location can be improved by triangulation of two or more visible APs (Access Points). The mobile client currently supports the application in automatically retrieving location information from nearby WiFi, BT and GSM location providers, and in interacting with the PDPT server. The application (locator) is implemented in C# language using the MS Visual Studio .NET with .NET Compact Framework and a special OpenNETCF library enhancement.

A first key step of the localization is a data collection phase. The information about the radio signals is recorded as a function of a user's location. The signal information is used to construct and validate models for signal propagation. Among other information, the signal strength (SS) is available where WiFi, BT and GSM networks are available.

To get a user position with more accuracy, the triangulation is currently used in PDPT framework. Other localization techniques like Monte Carlo localization can be used to get a better position if it is needed, but the PDPT framework provides good results only with triangulation techniques on a basic level of localization.

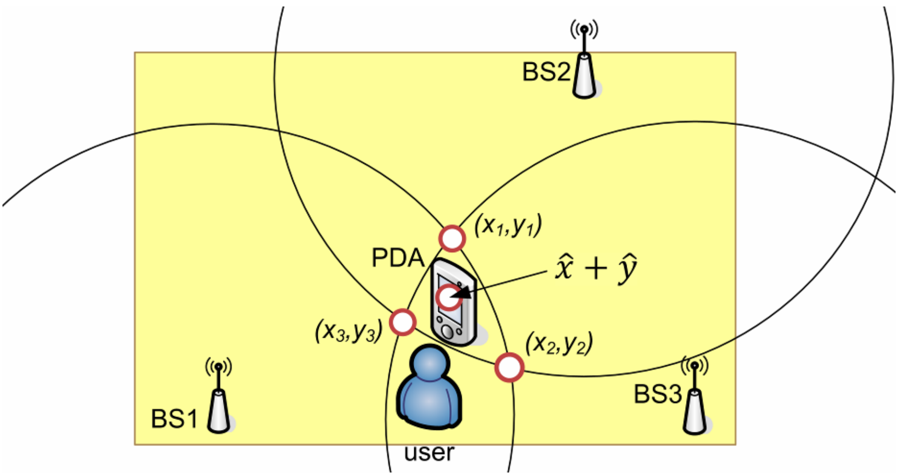

Fig. 12. Localization principle – trilateration

In a testing model (Fig. 12) the mobile client gets the SS info of three BSs (Base Stations) with some inaccuracy. Inaccuracy is caused by SS value from a mobile device wireless module, where only SS in present. Circles around the BSs (in real 3D space the sphere is used around the BSs representing SS value) are crossed in red points in the figure. The red point intersection (centre of three) is the best computed location of the mobile user. The user track is also computed from these locations and it is stored in a database for later use. This idea is applicable in the case of WiFi as well as BT and GSM networks.

4.1 WiFi Localization

In a real case of indoor localization by WiFi networks, several types of environments are used like open spaces, walls and mixed spaces. The Cisco APs (Cisco Aironet 1121 and 1131) are used in the test environment at the Technical University of Ostrava.

The measurements on three selected (representing three types of environment) APs of all APs have been performed to get signal strength (SS) characteristics. The characteristics were combined to get a one characteristic called “Super-Ideal characteristic”. The computed equation for Super-Ideal characteristic is taken as basic equation for PDPT Core to compute the real distance from WiFi SS. The equation has a sufficient coefficient of determination $R^2 = 80$ % (R^2= ssreg/sstotal).

In the case of in-door location the damping effect of walls especially when the number of BS´s is small could hamper the positioning. However the precise positioning is not needed in all cases. When the granularity of object areas to be prebuffered into the mobile device cache is in level less than tens of meters, the localization by one or two visible BS´s is possible with high level of success. Maximal location error for static localization is 25 meters for the case the only 1 WiFi AP is in the range, 7 meters for 2 APs in range, resp. less than 5 meters in case of 3 or more WiFi APs (mentioned Cisco models) in range. This localization error can be rapidly reduced by use of dynamic localization in a sense of user movement trajectory computation. Naturally, this localization principle can be applied to other wireless technologies like Bluetooth, GSM or WiMAX.

5 The PDPT Framework

The PDPT framework server is developed as a web service to act as a bridge between SQL Server (contain WLA database) and PDPT Clients (Fig. 13). Client mobile application contains a location sensor component to scan nearby for WiFi APs. WiFi SS info is continuously transferred to the PDPT Framework Core. This component computes the user's location information from WiFi SS. In next step the PDPT Core makes a decision to which part of WLA Server database needs to be replicated to client's SQL Server CE database [9][10]. The PDPT Core decisions constitute the most important part of PDPT framework, because the kernel must continually compute the position of the user and track, and predict his future movement. After doing this prediction the appropriate data are prebuffered to client's database for the future possible requirements. This data represent artifacts list of client buffer imaginary image.

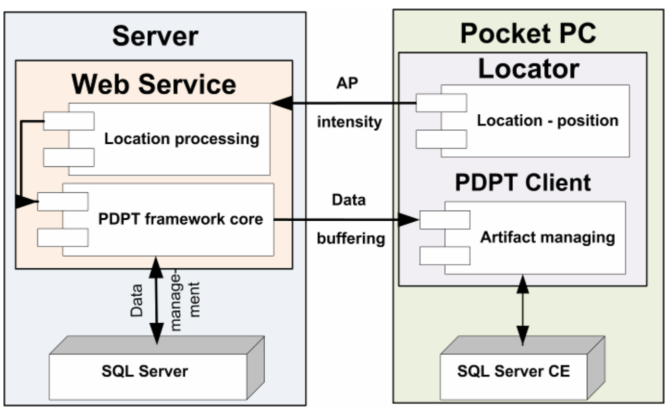

Fig. 13. PDPT architecture – UML design

5.1 Data Artifact Creating

Artifact represent an object in WLA SQL server database with image, audio, video or other file types. Every artifact must have associated with position coordinates in 3D environment (S-JTSK format is used). Open source software Quantum GIS is used to manage all data in 3D spaces like building map basis, APs location and artifacts location. To manage and work with locations of artifacts, firstly the building floor map is needed to obtain. In most cases the scanned version is adequate. The obtained map needs to be converted to Tagged Image File Format (TIFF). Location coordinates for such file must be created in TFW separate file. TFW file contains coordinates that describe the location, scale, and rotation of a map formatted as a raster TIFF image. All obtained position info must be stored in PDPT Core web service. Artifacts with position coordinates are stored in WLA database by “WLA Database Artifact Manager”.

5.2 Data Artifact Managing

The WLA server database manages the artifacts in the context of their location in building environment. The PDPT Core selecting the data to be copied from PDPT server to mobile client by context information (position info). Each database artifacts must be saved in database with the information about area to which it is belong.

A software application called “Data Artifacts Manager” was created to manage the artifacts in WLA database. User can set the priority, location, and other metadata of the artifact. The Manager allows creating a new artifact from multimedia file source (image, video, audio, etc.), and work with existing artifacts [9].

5.3 PDPT Core - Area Definition for Selecting Artifacts to Buffering

The PDPT buffering and predictive PDPT buffering principle consists of several following steps. Firstly the client must activate the PDPT on PDPT Client. This client creates a list of artifacts (Client buffer image), which are contained in his mobile SQL Server CE database. Server create own list of artifacts (imaginary image of Client buffer) based on area definition for actual user position and compare it with real Client buffer image. The area is defined as an object where the user position is in the center of object. The cuboid form is used in present time for initial PDPT buffering. This cuboid has a predefined area with a size of 10 x 10 x 3 (high) meters. The PDPT Core continues in next step with comparing of both images. In case of some difference, the rest artifacts are prebuffered to Client buffer. When all artifacts for current user position are in Client buffer, there is no difference between images. In such case the PDPT Core is going to make a predicted user position. On base of this new predicted user position it makes a new predictive enlarged imaginary image of Client buffer. The size of this new cuboid is predefined area of size 20 x 20 x 6 meters. The new cuboid has a center in direction of predicted user moving and includes a cuboid area for current user position. The PDPT Core compares the both new images (imaginary and real Client buffer) and it will continue with buffering of rest artifacts until they are same. Creation of an algorithm for dynamic area definition is better in real case of usage to adapt a system to user needs more flexible in real time [10].

6 Terain Identification and Navigation

The problem of orientating and navigating mobile devices in building divides into three options. The first option is orientation in a known place. It is assumed that the device is equipped with the map of the place. The only problem is to get the current position in relation with the map. The second option is the dynamic download of the necessary part of the map based on the current position – for example, the PDPT Framework. The third option is an unknown place. So the mobile system has to be able to dynamically recognize the environment and to dynamically generate the map. Then the mobile device has to find the way using some method.

The probe computes the current position based on four parts of information:

1. The probe approximates the position based on the WiFi connection or gets the position from GPS.
2. Map of the environment (stored or dynamically downloaded).
3. Clarifies the position in the map using sensors and possible positions.
4. Relative change of the position based on incremental sensors and accelerometers.

When the probe knows its current position it can reach the target position.

1. Using the map and Dijkstra's algorithm it will obtain the shortest path tree.
2. The probe first gets an item from the list – the current position to the first intersection.
3. Reach the intersection using the drive.
4. Check obstacles on the road.
5. Gets next branch from first intersection to second one.
6. If there is an obstacle turn left or right with 60 degrees based on space on sides of the road (sensor system).
7. Go back to previous direction – second intersection.
8. Repeat until target position is reached.

During the ride it is necessary to check the state of the embedded power source – the battery. Power consumption of the probe is displayed in (Fig. 14). During the ride the average power consumption of the probe is approximately 2.3 Ampers. The capacity of the battery is 4.6 Ah. So the probe can work in an active state for a maximum of 2 hours.

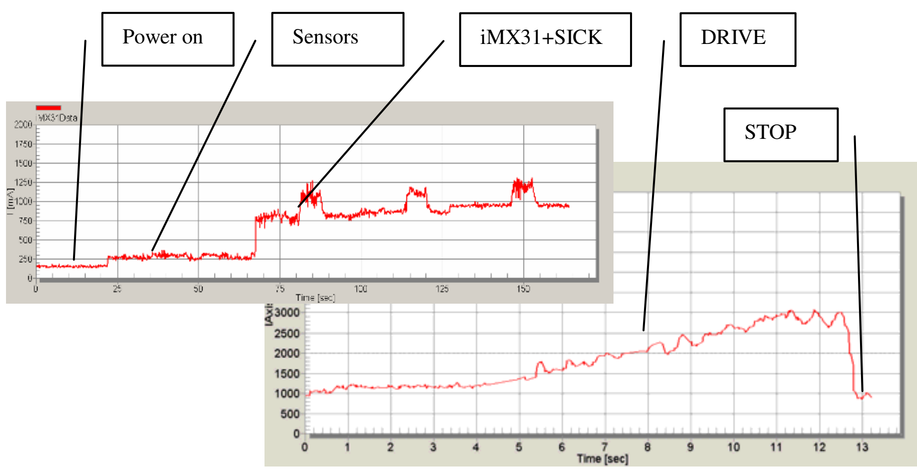

Fig. 14. Power consumption of the probe in five different states

The solar panel is prepared for charging the battery during the ride for the future. In the case of low energy the probe will stop, turn off the sensor system and wait to charge the battery.

7 Conclusion

The main aim of the project is to give the tool to the user which can extends the user perspective. The current state of the project is testing the set of mobile devices at the Technical University of Ostrava. The WiFi network is used for orientation in cooperation with local maps. The mobile device is equipped with an algorithm for finding the best way from the current to desired position. In the future the PDPT algorithm and dynamic generation of the map based on information from sensors will be implemented.

References

1. Abowd, G., Dey, A., Brown, P., et al.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, p. 304. Springer, Heidelberg (1999)
2. Krejcar, O.: User Localization for Intelligent Crisis Management. In: AIAI 2006, 3rd IFIP Conference on Artificial Intelligence Applications and Innovation, pp. 221-227 (2006)
3. Krejcar, O., Cernohorsky, J.: Database Prebuffering as a Way to Create a Mobile Control and Information System with Better Response Time. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 489–498. Springer, Heidelberg (2008)
4. Kotzian, J., Srovnal, V.: Distributed embedded system for ultralight airplane monitoring. In: sborníku (ed.) ICINCO 2007, Anger, France, vol. 2007/1, pp. 448–451 (2007), ISBN 978-972-8865-82-5
5. Kotzian, J., Srovnal, V.: Can Based Distributed Control System Modelling Using UML. In: Proceeding International Conference IEEE ICIT 2003, Maribor, Slovenia, pp. 1012–1017 (2003), ISBN 0-7803-7853-9
6. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
7. Haklay, M., Zafiri, A.: Usability engineering for GIS: learning from a screenshot. The Cartographic Journal 45(2), 87–97 (2008)
8. Evennou, F., Marx, F.: Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. Eurasip journal on applied signal processing (2006)
9. Krejcar, O.: Full Scale Software Support on Mobile Lightweight Devices by Utilization of all Types of Wireless Technologies. In: Granelli, F., Skianis, C., Chatzimisios, P., Xiao, Y., Redana, S. (eds.) Mobilight 2009. LNICST, vol. 13, pp. 173–184. Springer, Heidelberg (2009)
10. Krejcar, O.: Problem Solving of Low Data Throughput on Mobile Devices by Artefacts Prebuffering. EURASIP Journal on Wireless Communications and Networking, Article ID 802523, 8 pages (2010)
11. Timesys (online) (2010), LinuxLink, `https://linuxlink.timesys.com/3/Products` (cit. 2010-03-17)
12. Timesys (online) (2010), TimeStorm `https://linuxlink.timesys.com/3/Products/TimeStorm` (cit. 2010-03-17)
13. Zoom (online) (2010), Frescale Zoom i.MX31 LITEKIT, `http://www.logicpd.com/products/development-kits/freescale-zoom%E2%84%A2-imx31-litekit` (cit. 2010-03-17)
14. CiA (online), CAN in Automation (2010), `http://www.can-cia.org/` (cit. 2010-03-17)