A Pedestrian Navigation System for Car Locating

Mircea Popa*, Cristian Saracin,**
Anca Sorana Popa*, Marius Marcu******

**, **, ****Politehnica University from Timisoara, Faculty of Automation and Computers, Romania*
(e-mail: mircea.popa@ ac.upt.ro; cristian.saracin@ac.upt.ro; marius.marcu@ac.upt.ro)
**** Politehnica University from Timisoara, Faculty of Mechanics, Romania*
(e-mail: anca.popa@ mec.upt.ro)

Abstract: The data navigation systems became more and more present in current life. They can be divided in fixed (Internet based on desktops) and mobile. The mobile navigation systems make use of the tremendous development of smart mobile devices, such as PDAs, mobile phones or dedicated devices. They receive updated data through wireless Internet, through GPS and through specialized sensors. Car navigation systems are the most known implementation of data navigation systems. Pedestrian navigation systems (PNSs) were less focused. Although they have to respond to the same main requirement as car navigation systems, i.e. to guide a user from a starting point to a destination point, they have also different constraints. This paper presents a PNS for finding a car. It is intended to be mounted in the car's key and to guide the user from its starting point, which is in a delimited area around the car, to his car.

Keywords: GPS, PNS, magnetic sensor, navigation system, microcontroller.

1. INTRODUCTION

The data navigation systems became more and more present in current life. They can be divided in fixed (Internet based on desktops) and mobile. The mobile navigation systems make use of the tremendous development of smart mobile devices, such as PDAs, mobile phones or dedicated devices. They receive updated data through wireless Internet, through GPS and through specialized sensors.

Car navigation systems are the most known implementation of data navigation systems. They are embedded in cars or separated as distinct devices. Much attention was paid to those systems and their development was spectacular.

Pedestrian navigation systems (PNSs) were less focused. Certain types were developed, for example PNSs for visiting large delimitated areas (a museum or an institutional building) or PNS for monitoring the sportsmen during their training. But the most challenging task is to develop PNSs for guiding people in different areas, especially in metropolitan areas. Although they have to respond to the same main requirement as car navigation systems, i.e. to guide a user from a starting point to a destination point, they have also different constraints. Pedestrians' movement has a higher degree of freedom and they don't have to follow certain predefined ways. For example in order to traverse a wide open area a car has to follow certain routes while a pedestrian can move on the straight line from one corner to the opposite one. More than that, rich databases exist for a lot of street networks for many countries. They are used by car navigation systems but cannot be used by PNS as they are. Pedestrian navigation rarely happens exclusively on street networks. Pedestrian navigation systems can be used for tourist reasons or/and for finding a target, for example the own car previously left in a certain place.

This paper presents a PNS for finding a car. It is intended to be mounted in the car's key and to guide the user from its starting point, which is in delimited area around the car, to his car.

The rest of the paper is divided as follows: the next section presents related work, the third section describes the proposed system, the fourth section presents experimental results and the last section outlines the conclusions.

2. RELATED WORK

Pedestrian navigation systems are reported in different papers.

Reference [1] focuses on the implementation and deployment of a pedestrian navigation system which realizes a timely navigation by presenting landmark-based instruction of guidance using the spatial semantics stored locally.

In reference [2] a new pedestrian wayfinding model that addresses the problem of movement of the pedestrian under specific conditions is presented. A graph model is created which consists of decision points and edges connecting them. The free walkable space around decision points is expanded to decision scenes where pedestrian movement is modeled in more detail, thus allowing for flexible navigation comparable to unassisted pedestrians.

Reference [3] presents a system based on a low-cost inertial measurement unit and high-sensitivity global positioning system receivers for personal navigation in an environment where GPS signals are degrading. The system is mounted on a pedestrian shoe and uses measurements based on the dynamics experienced by the inertial sensors on the user's foot.

A positioning system for indoor pedestrian navigation services using mobile phones is described in reference [4]. It is intended for environments where GPS system can not be used. The system is made of smart phones and license-free radio beacon devices that can be driven with little electric power. There is no need for server-side computation because the system works autonomously, that is, the user's device receives wireless beacon signals from the environment and can detect a user's position independently from the mobile terminal.

The authors of reference [5] have developed a pedestrian navigation system that delivers photorealistic panoramic landscape images using 3D models of a city and related information to mobile phones. The requirements for smooth mobile phone navigation are presented (low processing load and quick data delivery) and a server-side panoramic image generation mechanism and a divided guidance information transfer technique.

Reference [6] discusses the use of active and passive landmarks, as well as multimedia presentation forms and general design goals for a combined indoor/ outdoor pedestrian navigation service. It is shown that landmarks are an important enhancement for pedestrian navigation systems but their diversity makes it difficult to include them in guiding instructions.

Reference [7] gives an overview of the newly developed ubiquitous positioning technologies and their integration in navigation systems. Two case studies are presented: the improvement of land vehicle safety using Augmented Reality technologies and pedestrian navigation services for guidance of users to certain university offices.

In reference [8] a pedestrian navigation system using texture paving blocks is presented. Colour texture paving blocks are used as markers. Positioning is performed by taking a picture of these markers with a mobile terminal built-in camera.

Unlike many other similar achievements, the solution from this paper has severe resource and energy consume constraints. It is intended to be implemented on an 8 bit microcontroller, with poor resources so the number of sensors must be minimized and the algorithms must be as simple as possible. Also, part of the above mentioned solutions use mobile phones or PDAs as mobile devices, having satisfactory energy resources and, most of all, rechargeable ones. This is not the case in the proposed PNS, were the key of the car has a low power non rechargeable battery. So, the energy consume must be as low as possible (measured values will be supplied in the fourth section).

3. THE PROPOSED SYSTEM

3.1 General Presentation

The main goal of the proposed PNS is to help the user to find his car starting from his current position. The system has two components: one for computing the direction the user must follow and one for displaying this direction in a friendly format. The system is mounted in the key of the car and has to harmonize with the already existent hardware and software.

Fig. 1 presents the sequence of operations which take place at the system and at the user.

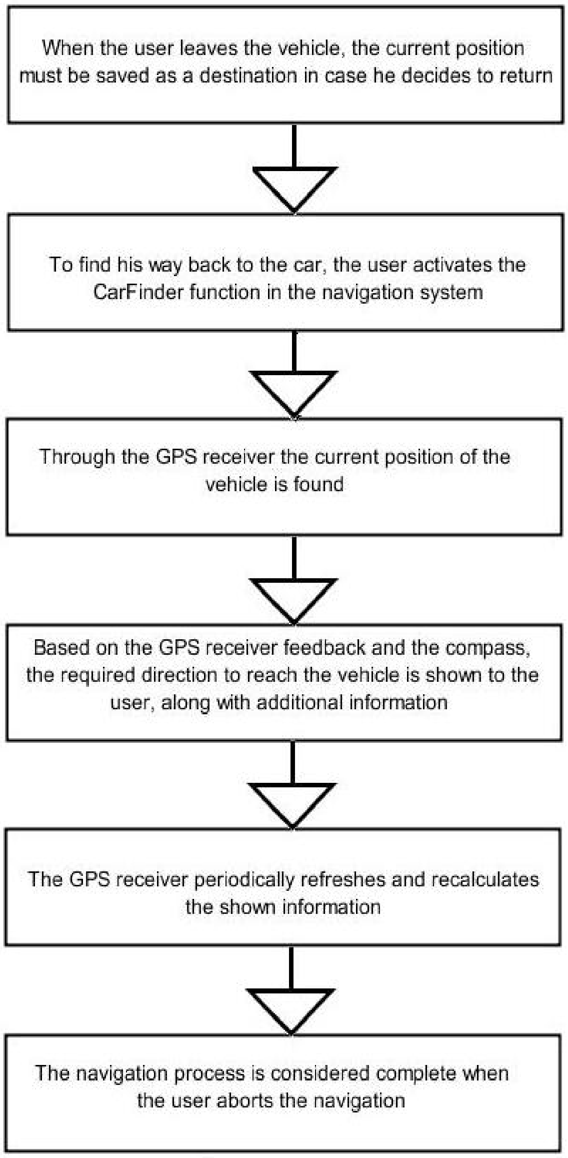

Fig.1. Sequence of operations at the interaction between the PNS and the user

The PNS has to function independently on the possible already existent car navigation system. The main drawback due to this constraint is that the PNS has to be active when the user leaves its car in order to memorize the car's position. Thus, the user must be close to the car until the PNS succeeds to obtain the current position. Dependent on the environment, this operation may last from several seconds to several minutes

The hardware of the PNS consists of a GPS receiver and a magnetic sensor. This hardware must be easily connected with the already existent hardware and due to the mobility requirement the minimization of the energy consume must be an important target.

The software of the PNS means the new navigation facilities which must be integrated in the existent software without degrading any of its performances.

3.2. The Hardware

A PNS is typically used when the user is pedestrian and has a slow movement, with smooth variations advised by a GPS receiver. This kind of system is much more vulnerable to the loss of GPS connectivity due to the medium in which the user moves. The user may be surrounded by tall buildings or may be inside of them, which raises the risk of loss of GPS connectivity. That is why the system must be provided with a supplementary navigation system, totally different from GPS. This new navigation system has to diminish the errors introduced by the GPS and to compensate the possible losses of GPS connectivity. A magnetic sensor was used in the proposed PNS. It indicates the current orientation of the user in the N-E-S-W axes system. For that, the magnetic sensor reads the values of the magnetic field of the Earth on the X, Y, Z axes. These readings are given to a microcontroller which transforms them in a bidimensional representation of the position of the user and guides him to the destination. Fig. 2 presents a block diagram of the hardware of the PNS.

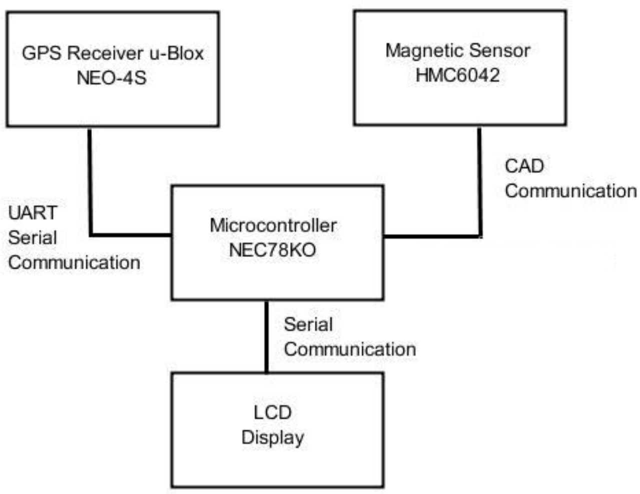

Fig.2. Block diagram of the hardware of the PNS

The uBlox NEO-4S GPS receiver and the Honeywell HMC6042 magnetic sensor were used. The GPS receiver is connected to the UART interface of a microcontroller and the outputs of the magnetic sensor are connected to the ADC inputs of the microcontroller. The GPS receiver and the magnetic sensor are implemented as a separate module which is connected to the already existent hardware of the access key through a 9 pin connector. The existing hardware fulfils all the required functions for a classical access key of a car. It consists in a RF module for wirelessly commanding the car, a keyboard for opening/ closing the doors and for alarming functions, a displaying module consisting in a LCD display and a led and a battery based power supply module. All these modules are commanded by an 8 bit NEC78K0 microcontroller, the same microcontroller which commands the other modules of the key in order to offer the classical functions of a car access key.

The main features of the microcontroller are: 32 8 bit Special Function Registers, internal Flash memory, watchdog timer, 41 standard input/ output ports, 2 temporizers/ counters for counting internal clock pulses or external events, several serial interfaces: 2 UARTs, CSI and I^2C, 8 channel analog/ digital converter with 10 bit resolution, power supply: 1.8 – 5.5 V.

Next, several details will be given concerning the two modules of the microcontroller used by the GPS receiver and the magnetic sensor. The UART6 interface ensures an asynchronous serial communication with the following features: maximum rate transfer: 312.5 kbps, 7 or 8 bit data format, an internal baud generator, thus allowing any value for the transfer rate, full duplex communication, data can be sent in MSB or LSB configuration.

The ADC converts analogical inputs in a 10 bit digital format. There are 8 analogical inputs and when a conversion took place, the ADC generates an interrupt request. The input can be selected by program and the conversion process can be started either by hardware or by software.

3.3. The Software

Fig. 3 presents the block diagram of the software part of the PNS and its interactions with the already existing software.

The software of the PNS is divided in the following parts: the magnetic sensor driver, the GPS receiver driver, the computing module, the LCD computing module and the application module.

The magnetic sensor driver implements the communication with the HMC6042 sensor. It ensures:

- initialization of the communication with the magnetic sensor: it means the configuration of the microcontroller's ADC;
- connecting and disconnecting the magnetic sensor to the power supply;
- providing the values of the intensity of the magnetic field: the values are obtained by converting the readings of the magnetic sensor;

The GPS receiver driver implements the communication with the uBlox NEO-4S GPS receiver. It ensures:

- initialization of the communication with the GPS

receiver: it means the configuration of the microcontroller's asynchronous serial interface;
- activation of the GPS receiver: it means the wake-up of the receiver in order to provide the positioning information;
- reception of the information from the GPS receiver, their verification and their transmission to the upper software layers;
- disconnecting of the GPS receiver: putting the GPS receiver in its low-power mode.

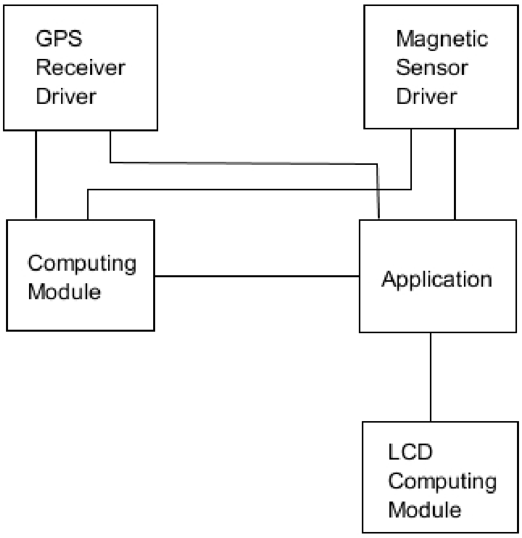

Fig. 3. Block diagram of the software of the PNS

The computing module achieves all the needed computations in order to obtain the direction the user must follow and the distance until the destination. The direction is obtained from the data provided by the GPS receiver and the magnetic sensor. The direction is a function of the following parameters:
- azimuth: it is the orientation against the magnetic poles of the Earth;
- bearing: it is the angle against the north between the line which connects the current position and the car's position (it gives the movement's sense).

The computing module ensures:
- the calculus of the azimuth: the value is obtained from the data provided by the magnetic sensor;
- the calculus of the direction: this is obtained from the current position and the car's position;
- the calculus of the distance: it is also obtained from the current position and the car's position.

The LCD computing module displays the information on the LCD display.

The application implements the interaction between the user and the PNS. The following services exist:
- implementation of the navigation-specific interactions: saving of the current position and navigation towards the car;
- implementation of the user-specific interactions: interrogation of the GPS driver for obtaining the current position, interrogation of the magnetic sensor driver for obtaining the current values of the intensity of the magnetic field;
- interrogation of the computing module for obtaining the direction and the distance to the car;
- displaying the information necessary to the user on the LCD already existent.

Next, implementation details concerning the computing module will be provided. It contains functions for converting the latitude and the longitude from the GPS format in degrees and functions for calculating the bearing value, the azimuth value, the direction and the distance towards the destination from the current position.

The function `gpsCalc_convertLatitude` converts the string of characters received from the GPS receiver and memorized in the microcontroller's internal memory in a value in degrees, the range being (0, 90). The string of characters has the form ggmm.mmmmS, i.e. GradesMinutes, FractionsofMinutesSense. The conversion is done according with the following formulas:

$$D = Grades + \frac{Minutes}{60} + FractionsofMinutes * \frac{0.06}{3600}$$

for Sense = N and

$$D = -(Grades + \frac{Minutes}{60} + FractionsofMinutes * \frac{0.06}{3600})$$

for Sense = S.

The sequence is:

```
double        gpsCalc_convertLatitude(char
*Latitude){
       int dumm ;
       int l_d,l_m,l_s ;
       char dir ;
       sscanf (Latitude,"%2d%2d.%3d%2d%c",
&l_d,&l_m,&l_s,&dum,&dir);
       return returnDegrees(l_d,l_m,l_s,di
r);
}
```

The function `gpsCalc_convertLongitude` converts the string of characters received from the GPS receiver and memorized in the microcontroller's internal memory in a value in degrees, the range being (0, 90). The string of characters has the form ggmm.mmmmS, i.e. GradesMinutes, FractionofMinutesSense. The conversion is done according to the same formulas as the ones already presented, but for Sense = E, respectively for Sense = W. The sequence is:

```
double      gpsCalc_convertLongitude(char
*Longitude){
       int dumm ;
       int l_d,l_m,l_s ;
       char dir ;
       sscanf(Longitude,"%3d%2d.%3d%2d%c",
&l_d,&l_m,&l_s,&dum,&dir);
       return returnDegrees(l_d,l_m,l_s,di
r);
}
```

The bearing value is calculated according to the following formula:

$$bearing = arctg2(\sin(\Delta long)\cos(lat2), \cos(lat1)\sin(lat2) - \sin(lat1)\cos(lat2)\cos(\Delta long))$$

$$\Delta long = long2 - long1$$

where lat2 and long2 are the coordinates of the car and lat1 and long1 are the current position coordinates.

The `gpsCalc_getBearing` function returns the bearing value. The sequence is:

```
double gpsCalc_getBearing(double lat_deg1, double long_deg1, double lat_deg2, double long_deg2){
//get the bearing between [lat1,long1] and [lat2,long2]
	double bearing;
	lat_deg1=lat_deg1*_PI/180;
	lat_deg2=lat_deg2*_PI/180;
	long_deg1=long_deg1*_PI/180;
	long_deg2=long_deg2*_PI/180;
	double y=sin(long_deg2-long_deg1)*cos(lat_deg2);
	double
x=cos(lat_deg1)*sin(lat_deg2)-sin(lat_deg1)*cos(lat_deg2)*cos(long_deg2-long_deg1);
	bearing=atan2(y,x);
	bearing=bearing*180/_PI;
return bearing;
}
```

The azimuth value is calculated using the data obtained from the magnetic sensor. The sensor must be calibrated for reducing the effects of the external magnetic interferences due to the metallic objects found close to the sensor. The calibration is done only in the manufacturing phase, it will not be necessary during the regular functioning of the system. The calibration uses the minimum and maximum values, Xmin, Xmax, Ymin and Ymax, read by the magnetic sensor on the two axes when rotating it in the direction N-E-S-W and other two values, on each axe, the offset values, Xoff and Yoff and the scaling factor, Xsf and Ysf.The formulas used for calculating the azimuth are:

$$Xsf = \max(1, \frac{Y\max - Y\min}{X\max - X\min})$$

$$Ysf = \max(1, \frac{X\max - X\min}{Y\max - Y\min})$$

$$Xoff = (\frac{X\max - X\min}{2} - X\max) * Xsf$$

$$Yoff = (\frac{Y\max - Y\min}{2} - Y\max) * Ysf$$

The values Xd, Yd provided by the magnetic sensor are used for calculating the intensity of the magnetic field:

$$X = Xd * Xsf - Xoff, Y = Yd * Ysf - Yoff$$

Fig. 4 presents the algorithm implemented for calculating the azimuth.

The function `gpsCalc_getAzimuth` receives the values for the intensity of the magnetic field read by the magnetic sensor and returns the value of the azimuth according to the above algorithm. The sequence is:

```
double gpsCalc_getAzimuth(unsigned int conversionResultX, unsigned int conversionResultY){
	int x=(conversionResultX>>6)*XSF-abs(XOFF);
	int y=(conversionResultY>>6)*YSF-abs(YOFF);
	double
	t=atan(((double)y/(double)x);
		t=t*180/_PI ;
		if (x==0 && y<0) return 90;
		if (x==0 && y>0) return 270;
		if (x>0 && y>0) return (int)(360-t);
		if (x>0 && y<0) return (int)(t*-1);
		if (x<0)
		return (int)(180-t);
		return 0;
}
```

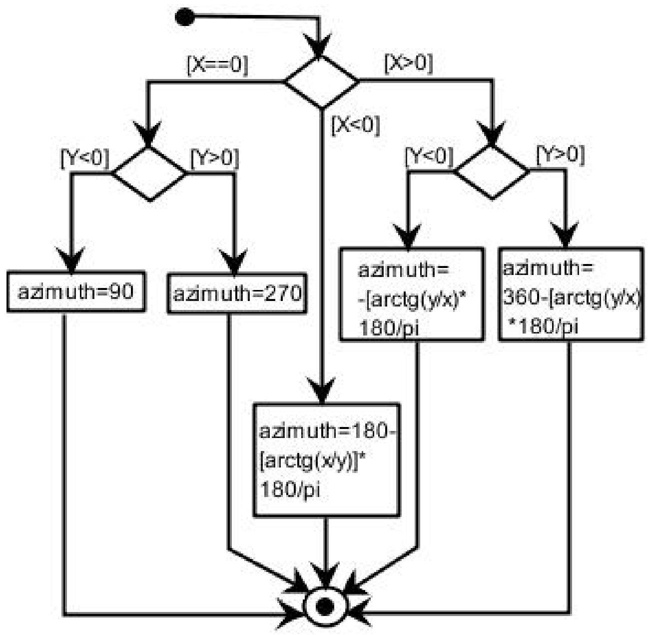

Fig. 4. The algorithm for the azimuth value

The function `gpsCalc_getHeading` receives the values for the azimuth and the bearing and returns the heading value which indicates the direction.

Fig. 5 presents the algorithm used for calculating the direction (heading value).

3.4 The User Interface

Through the LCD display, the PNS offers the following services: Current position save, Start navigation and Stop navigation.

Current position save: by activating this option, the system saves the current position provided by the GPS receiver. Data are memorized in the internal Flash memory of the microcontroller. The position is represented by the pair (latitude, longitude).

Start navigation: when this option is activated, the system will provide to the user information concerning the position of the car. The information available to the user is: the direction to the car and the distance to it. The information is periodically actualized by the GPS receiver and the magnetic sensor. The direction is shown by an arrow on the LCD display. 16 possible positions are shown leading to a 22.5 degree resolution. A higher resolution might lead to a higher sensitivity to external perturbation factors resulting erroneous information. Fig. 6 shows how the direction is displayed.

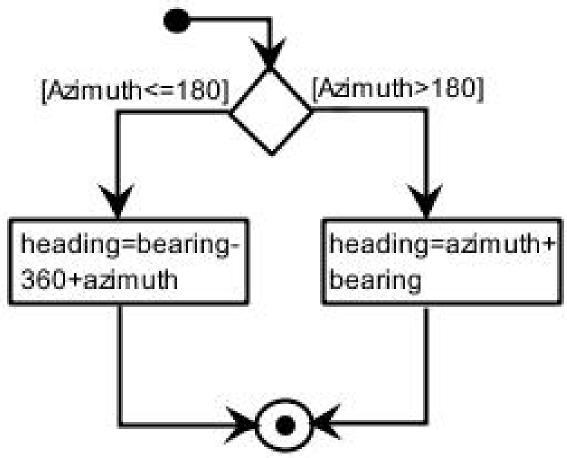

Fig. 5. The algorithm for the heading value

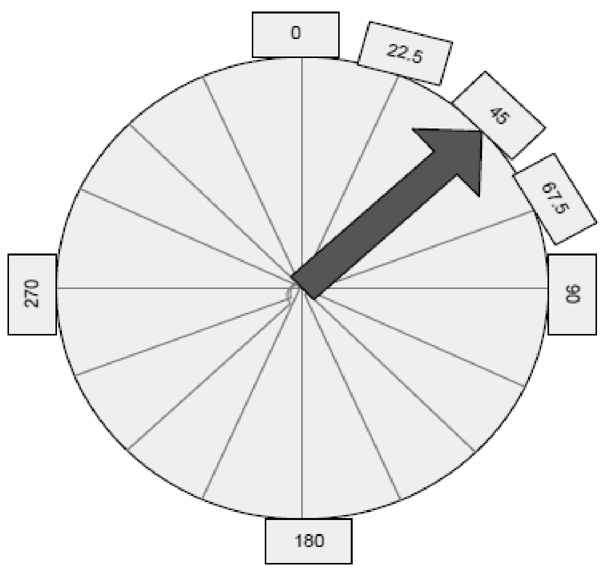

Fig. 6. Showing the direction on the PNS's LCD display

Stop navigation: the GPS receiver and the magnetic sensor are switched to low consume modes.

4. EXPERIMENTAL RESULTS

The hardware under test is composed by: the uBlox NEO-4S GPS receiver with the ANN-MS antenna and the HMC6042 magnetic sensor connected to a NEC78K0 microcontroller. The data collected by the microcontroller are sent to a PC through an asynchronous serial interface. All the computations necessary for navigation are done on the PC. With this hardware, a route is achieved. The purpose of the test is to verify the accuracy of the data provided by the GPS receiver and to verify the validity of the algorithm for computing the direction and the distance. The starting strategy of the GPS is Cold start.

The experiments were done in the local town. A real route and a calculated one were obtained using the indications of the pedestrian system. The destination point has the following real coordinates: 45.740051^0 latitude and 21.238951^0 longitude. As it will be shown, there is a slight difference from the calculated coordinates. In order to show the routes, photos from GoogleMaps were used. The photos from GoogleMaps have a certain rate of update (usual once/year) but this is not a disadvantage here because their role is only to serve as a support for showing the obtained routes.

Table 1 shows measured values for the energy consume of the proposed PNS, in different cases.

Table 1. Energy consume of the proposed PNS

Key components	Display	GPS receiver	Magnetic sensor	Peak current consumption [mA]
Power off	Inactive	Standby	Power off	1.5
Active	Active	Standby	Power off	35
Active	Active	Cold start	Active	95
Active	Active	Continuous tracking	Active	80

In the first set of experiments the route is in an area less friendly, with a lower visibility for the GPS Fig. 7 presents the real route and the reconstituted one, based on the information given by the GPS receiver. Due to the loss of visibility there are differences between the two routes. Fig. 8 presents the variation of the distance to the destination and fig. 9 shows the variation of the HDOP (Horizontal Dilution of Precision), received from the GPS receiver, parameter which gives the precision with which the GPS receiver has made the calculus.

Fig. 7. Real and reconstituted routes for experiment 1

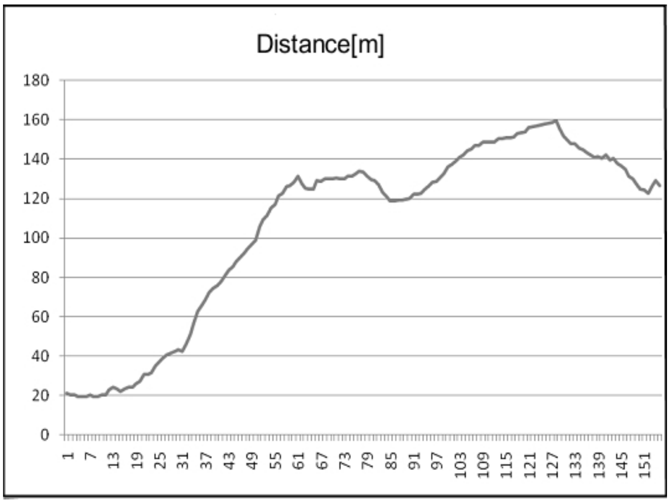

Fig. 8. Variation of the distance to the destination for experiment 1

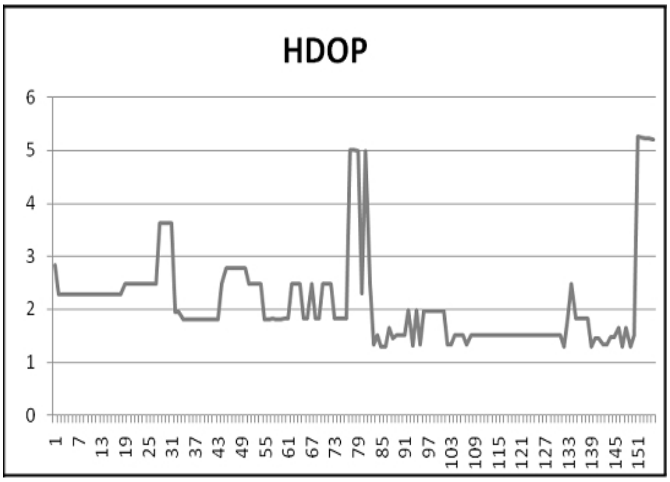

Fig. 9. Variation of HDOP for experiment 2

If the values for distance, bearing and coordinates of the current position are known, one can calculate the coordinates of the destination point using the following formulas:

$$lat2 = \arcsin(\sin(lat1) * \cos\frac{D}{R} + \cos(lat1) * \sin\frac{D}{R} * \cos(Bearing))$$

$$long2 = long1 + arctg2(\sin(Bearing) * \sin\frac{D}{R} * \cos(lat1), \cos\frac{D}{R} - \sin(lat1) * \sin(lat2))$$

The above formulas introduce a systematic error, of 0.000095 m. This error represents the distance between the destination with calculated coordinates and the destination with real coordinates. For example, if the coordinates of a certain current point (from fig. 7) are 45.73974^0 latitude and 21.23917^0 longitude, the calculated coordinates of the destination are 21.238951^0 longitude and 45.740052^0 latitude and they differ from the real coordinates of the destination (21.238951^0 longitude and 45.740051^0 latitude). The distance between the two points, the calculated destination and the real destination is 0.000095 m.

In the second set of experiments, the previous conditions are the same, except the environment has a very low visibility for the GPS receiver with a portion having a better visibility. The time needed for Cold start was 150 sec. Fig. 10 presents the real route and the reconstituted one, fig. 11 presents the variation of the distance to the destination and fig. 12 presents the variation of the HDOP parameter. One can observe significant differences between the real route and the reconstituted one and high errors. The differences and the errors diminish considerably when the GPS reaches the zone with better visibility.

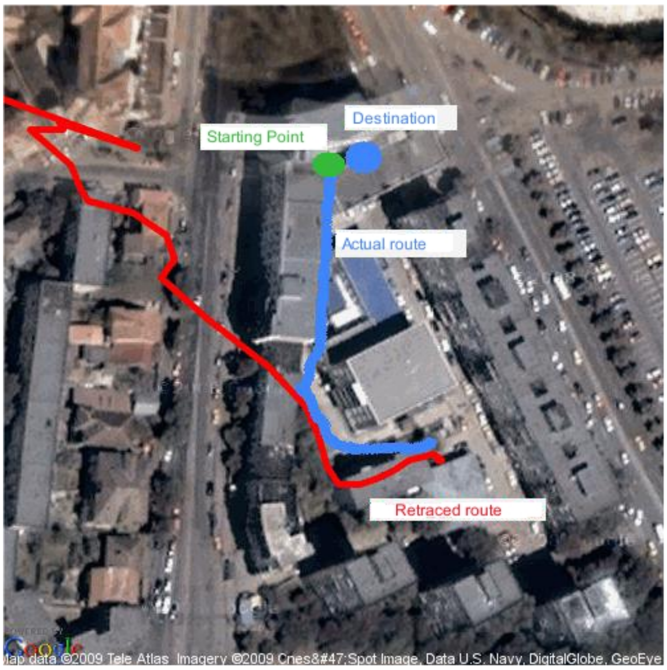

Fig. 10. Real and reconstituted routes for experiment 2

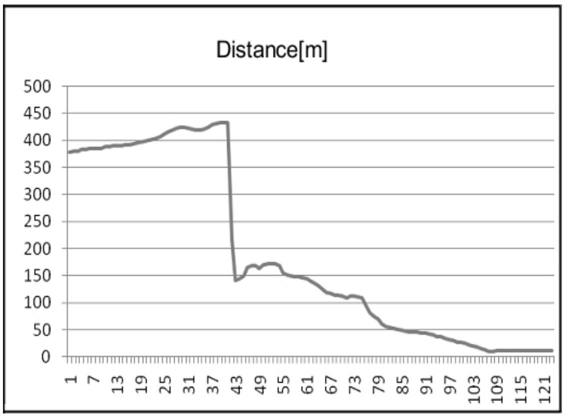

Fig. 11. Variation of the distance to the destination for experiment 2

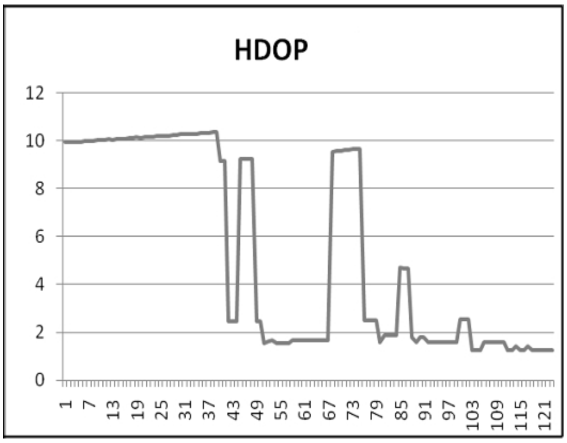

Fig. 12. Variation of HDOP for experiment 2

In the third set of experiments the environment was with high visibility for the GPS receiver. The starting time was only 50 sec. Fig. 13 presents the real route and the reconstituted one, fig. 14 presents the variation of the distance to the destination and fig. 15 presents the variation of the HDOP parameter. One can observe small differences between the routes and rather small variations of the HDOP parameter.

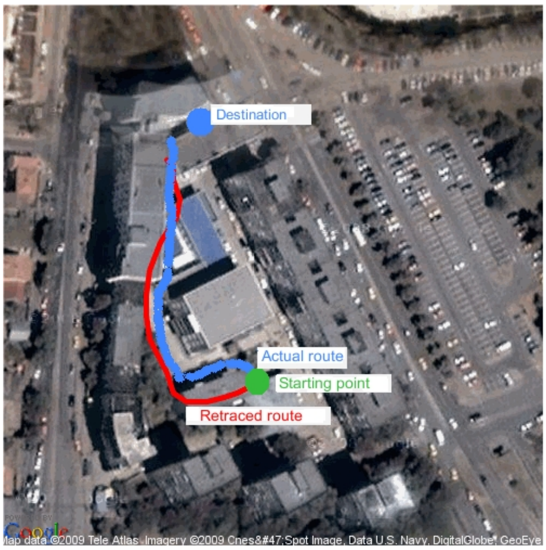

Fig. 13. Real and reconstituted routes for experiment 3

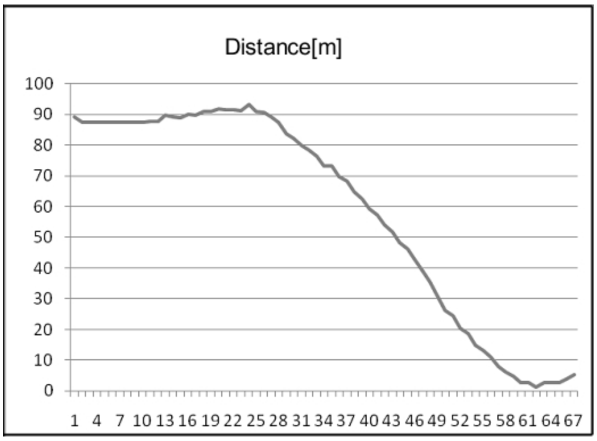

Fig. 14. Variation of the distance to the destination for experiment 3

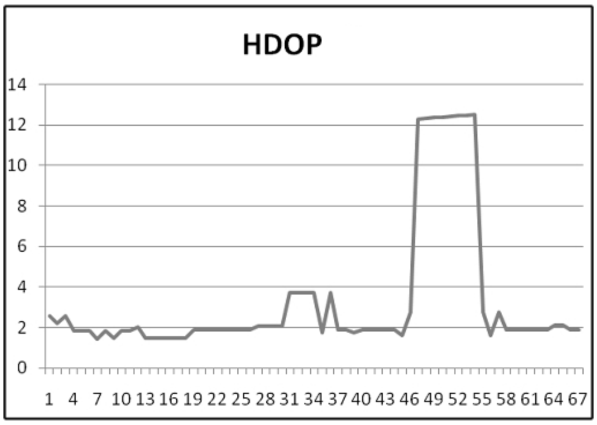

Fig. 15. Variation of HDOP for experiment 3

5. CONCLUSIONS

The paper has presented a pedestrian navigation system intended to find a car. The system is mounted in the access key and guides the owner from a starting point to his car.

Further development directions are:

- the replacement of the magnetic sensor with a more sensitive one; this will save the calibration operations which are necessary now;
- the improvement of the tracking algorithms; the idea is to save in the system the routes the user has followed and when the guiding system is activated to use the saved routes, avoiding the time and the energy needed to construct new ones.

ACKNOWLEDGMENT

This work was supported by Romanian Ministry of Education, CNCSIS grant 680/19.01.2009.

REFERENCES

Bessho M., Kobayashi S., Koshizuka N., Sakamura K (2008) uNavi: Implementation and Deployment of a Place-Based Pedestrian Navigation System, in Proc. of the 32nd Annual IEEE International Computer Software and

Applications Conference (COMPSAC 2008), Turku, Finland

Gaisbauer C., Frank A.U (2008) Wayfinding Model For Pedestrian Navigation, in Proc. of 11th AGILE International Conference on Geographic Information Science, University of Girona, Spain

Godha S., Lachapelle G (2008) Foot mounted inertial system for pedestrian navigation, in Measurement Science and Technology Journal, Issue 7, July 2008

Inoue Y., Sashima A., Kurumatani K (2009) Indoor Positioning System Using Beacon Devices for Practical Pedestrian navigation on Mobile Phone, in. Lecture Notes In Computer Science, Vol. 5585, Proc. of the 6th International Conference on Ubiquitous Intelligence and Computing, Brisbane, Australia, pp. 251-265

Miyazaki Y., Kamiya T (2006) Pedestrian Navigation System for Mobile Phones Using Panoramic Landscape Images, in Proc. of the 2006 International Symposium on Applications and the Internet (SAINT'06), Phoenix, Arizona

Radoczky V (2007) How to design a pedestrian navigation system for indoor and outdoor environments, in Gartner G., Cartwright W, Peterson M.P (ed) Location Based Sevices and TeleCartography, Springer Berlin Heidelberg, ISBN 978-3-54-36727-7, pp. 301-316

Retscher G., Kealy A (2005) Ubiquitous Positioning Technologies for Intelligent Navigation System, in Proc. of The 2nd WORKSHOP ON POSITIONING, NAVIGATION AND COMMUNICATION (WPNC'05) & 1st ULTRA-WIDEBAND EXPERT TALK (UET'05), Hannover, Germany, pp. 99-108

Yamashita S., Hasegawa T (2005) On the M-CubiTS pedestrian navigation system using texture blocks and its experiments, in Proc. of the 2005 IEEE Intelligent Transportation Systems, Vienna, Austria, pp. 121-125