INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 5, Issue 5 OCT 2016

NOVEL DEPLOYMENT SCHEMES FOR TARGET COVERAGE AND NETWORK CONNECTIVITY IN MOBILE SENSOR NETWORKS

CH.SWATHI [1,] **G.D.BASAVARAJ** [2]

8swathiguptha@gmail.com[1] gdbraj@gmail.com[2]

[1] PG Scholar, Dept Of ECE Sreyas Institute of engineering and technology Tatti annaram (v), Hayathnagar Mandal, Rangareddy Dist, Telangana, India.

[2] Guide, M.Tech, Assistant professor, Dept Of ECE Sreyas Institute of engineering and technology, Tatti annaram (v), Hayathnagar Mandal, Rangareddy Dist, Telangana, India.

Abstract Coverage of interest points and network connectivity are two main challenging and practically important issues of Wireless Sensor Networks (WSNs). Wireless sensor network is a group of physically distributed sensor nodes. Sensors nodes can be mobile or static. In mobile sensor network, the sensors have the capacity of mobility for better make use of resources, relocations and for better network lifetime. This movement of sensors consumes much more energy than sensing the environment. The sensor movement has to be minimized meanwhile covering the target and also network connectivity between the sensor nodes. Efficient solutions for target coverage and network connectivity are proposed in which more attention is given to minimization of sensor movement. The proposed method track the inertial movements of a moving object and to show effective results in a graphical manner in mat lab and in android mobile using WSN

Keywords—COMPONENTS, ARM BOARD, MEMS SENSOR, COMPASS SENSOR

Introduction

In recent years, Wireless Sensor Network (WSN) has arisen as efficient technology for wide range of applications. The applications of wireless sensor network include environmental monitoring, object tracking, traffic management, emergency navigation, etc. A wireless sensor network is a set of physically distributed sensor nodes. Sensor node is a small wireless device with limited battery life, radio transmission range and storage size. A sensor node performs the task of collecting important data, processing the data, monitoring the environment, etc. Sensor nodes in the network communicate with each other using radio transmitter and receiver. Generally sensor nodes have three units: Sensing unit, Communication unit and processing unit. Sensor node collects relevant data from the environment send it to the sink node or base station via single hop or multi hop communication. Base station is the central authority in the network. Base station has ability to monitor the sensor nodes. At the base station data is aggregated. Sensor nodes can be mobile or static. Mobile Sensor network is the group of moving sensor nodes. Mobile sensor networks have additional capacity of Mobility. Mobility consists of different functions in sensor network like better network lifetime, better use of resources, relocation, etc. Sensor nodes may change their location after initial deployment. Mobility can apply to all nodes or only to subgroups of nodes. Mobility can be active or passive. In active mobility the sensors are able to find their path and move while in passive sensors they may be moved by human or environmental assistance. Mobility of the sensor nodes can affect the overall performance of the network. Sensor deployment is another issue in mobile sensor networks, because it not only determines the cost of constructing the network but also affects how well a location is monitored by a sensor node. Sensor deployment can affect the quality of coverage and connectivity. Target coverage and Network connectivity are two major issues of Mobile sensor networks. Target coverage covers a set of interested points in deployment area of mobile sensor networks. It guarantees that every target is covered by at least one mobile sensor. Network Connectivity guarantees that there must be sufficient routing paths between sensors. Target coverage is affected by a sensor's sensing range, whereas Network connectivity is decided by a sensor's communication range. Target coverage and network connectivity may also affect the performance of Network.

Ease of use

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious. All embedded systems are Page 1765 including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer. The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware.

Arm processor review: ARM stands for Advanced RISC Machines. It is a 32 bit processor core, used for high end application. It is widely used in Advanced Robotic Applications. It performs number of instruction in a single cycle compare with other controllers it have advanced features. The Arm CPU with real-time emulation and embedded trace support that combine microcontroller with embedded high speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and a unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30 % with minimal performance penalty. Due to their tiny size and low power consumption, LPC2141/42/44/46/48 are ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale. Serial communications interfaces ranging from a USB 2.0 Full-speed device, multiple UARTs, SPI, SSP to I2C-bus and on-chip SRAM of 8 kB up to 40 kB, make these devices very well suited for communication gateways and protocol converters, soft modems, voice recognition and low end imaging, providing both large buffer size and high processing power.

History and Development:

ARM was developed at Acron Computers ltd of Cambridge, England between 1983 and 1985.

RISC concept was introduced in 1980 at Stanford and Berkley. ARM ltd was found in 1990.

ARM cores are licensed to partners so as to develop and fabricate new microcontrollers around same processor cores.

Key features:

1.16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.

2. 8 kB to 40 kB of on-chip static RAM and 32 kB to 512 kB of on-chip flash memory. 128-bit wide interface/accelerator enables highspeed 60 MHz operation.

3. In- System Programming/In-Application programming (ISP/IAP) via on-chip boot loader software. Single flash sector or full chip erase in 400 ms and programming of 256 bytes in 1 ms.

Related work: The use of mobile devices for gathering traffic information is not a new concept; several works indicate the feasibility of an ITS based only on location samples gathered by mobile phones. An early work describes an analytical method for evaluating real-time ITS based on data collected from GPS devices in probe vehicles: a 3-5% of penetration in the traffic flow is enough for adequate traffic estimation. Recent experiments with a system implemented solely on mobile phones show encouraging results for the feasibility and the accuracy of the traffic estimation (compared to that obtained by fixed sensors): a 2-3% penetration of mobile phones running the application in the total car flow suffices for accurate estimation of the average speed. Moreover, commercial navigation applications already integrate location samples from mobile phones in their algorithms for route guidance. However, security and privacy of similar traffic systems remain open challenges and research is conducted in several projects. Successive location updates by a smart phone, even without any identifier, contain spatial and temporal correlation that can be used as indirect identifiers. These can be exploited to reconstruct user paths with tracking techniques. Then traces can be processed and matched in order to infer frequently visited places, e.g., home or workplace, and finally reveal the user identity. To mitigate such threats, several solutions using cloaking techniques or privacy preserving sampling techniques have been proposed. These solutions are complementary to our proposal. In this paper we do not consider this kind of threat against the dataset of location samples. Rather, our goal is to guarantee the anonymity of the location samples and protect the system security. Relevant research in security is conducted for vehicular communication systems .Multiple short-term anonymized certificates, termed pseudonyms, can provide authentication while enhancing location privacy. These certificates are used for a short time and then have to be changed .Group signatures are also proposed, in order to reduce the overhead of pseudonym management. As they are significantly costlier (in terms of communication and computation overhead) than classic public key cryptography, special care must be taken for the overall secure vehicular communications system design. Group signatures are also used in credentials systems such as Idemix that provide anonymity for authenticated transactions to services. In our proposed architecture we will use group signatures; based on initial implementation results.

Proposed System

In this proposed block diagram consists of two parts one is Hardware and another one is Software.

In this Hardware part consists of two sensor of MEMS and Compass sensor along with Microcontroller, LCD Bluetooth module.

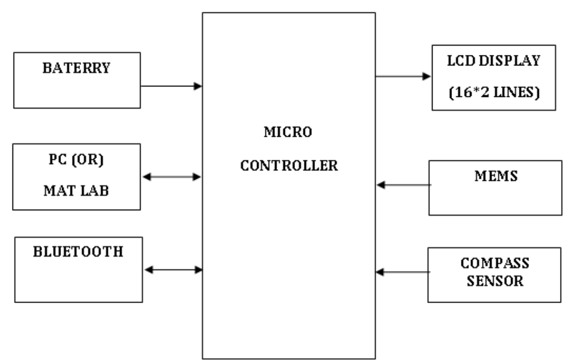

Fig.1.Block diagram

In this software part we are using Keil software. Keil software It is possible to create the source files in a text editor such as Notepad, run the Compiler on each C source file, specifying a list of controls, run the Assembler on each Assembler source file, specifying another list of controls, run either the Library Manager or Linker (again specifying a list of controls) and finally running the Object-HEX Converter to convert the Linker output file to an Intel

Hex File. Once that has been completed the Hex File can be downloaded to the target hardware and debugged. Alternatively KEIL can be used to create source files; automatically compile, link and covert using options set with an easy to use user interface and finally simulate or perform debugging on the hardware with access to C variables and memory. Unless you have to use the tolls on the command line, the choice is clear. KEIL Greatly simplifies the process of creating and testing an embedded application.

Projects The user of KEIL centers on "projects". A project is a list of all the source files required to build a single application, all the tool options which specify exactly how to build the application, and – if required – how the application should be simulated. A project contains enough information to take a set of source files and generate exactly the binary code required for the application. Because of the high degree of flexibility required from the tools, there are many options that can be set to configure the tools to operate in a specific manner. It would be tedious to have to set these options up every time the application is being built; therefore they are stored in a project file. Loading the project file into KEIL informs KEIL which source files are required, where they are, and how to configure the tools in the correct way. KEIL can then execute each tool with the correct options. It is also possible to create new projects in KEIL. Source files are added to the project and the tool options are set as required. The project can then be saved to preserve the settings. The project is reloaded and the simulator or debugger started, all the desired windows are opened. KEIL project files have the extension Simulator/Debugger **The simulator/ debugger** in KEIL can perform a very detailed simulation of a micro controller along with external signals. It is possible to view the precise execution time of a single assembly instruction, or a single line of C code, all the way up to the entire application, simply by entering the crystal frequency. A window can be opened for each peripheral on the device, showing the state of the peripheral. This enables quick trouble shooting of mis-configured peripherals. Breakpoints may be set on either assembly instructions or lines of C code, and execution may be stepped through one instruction or C line at a time. The contents of all the memory areas may be viewed along with ability to find specific variables. In addition the registers may be viewed allowing a detailed view of what the microcontroller is doing at any point in time. The Keil Software ARM development tools listed below are the programs you use to compile your C code, assemble your assembler source files, link your program together, create HEX files, and debug your target program. µVision4 for Windows Integrated Development Environment: combines Project Management, Source Code Editing, and Program Debugging in one powerful environment.

What's New in µVision4? µVision4 adds many new features to the Editor like Text Templates, Quick Function Navigation, and Syntax Coloring with brace high lighting Configuration Wizard for dialog based startup and debugger setup. µVision4 is fully compatible to µVision3 and can be used in parallel with µVision3.

Advantages

- Ease of controlling.
- Fast response.
- The module can be made into various forms as per the area of application.
- User friendly- One need not to know about the robot, as they can control by hand movement.

Conclusion and future scope

In this paper, we have presented a new approach toward the development of a gesture-based human–machine interface. An end-to-end approach is presented which maps arm-scale gesture by a human user to a learned response by a robotic agent through repeated applications of user-provided reward. Between these two end points, the constituent challenges are addressed in the areas of sensor selection, data representation, pattern recognition, and machine learning. As a composite approach, the proposed system overcomes many of the shortcomings of previous efforts. This project proposes an integrated approach to real time detection, tracking and direction recognition of sensor movements, which is intended to be used as a human-gesture interaction to demonstrate inertial movements. Integrating a single chip wireless solution with a MEMS accelerometer would yield an autonomous device, small enough to apply to the system, because of their small size and weight. Accelerometers are attached to any part of the system. Arrows on the hand show the location of accelerometers and their sensitive directions. That the sensitive direction of the accelerometer in the plane of the hand. Through accelerometer the object's inertial movements and compass sensor conditions are taken by the controller and is given to Mat Lab for graphical representation, the same data can be also represented in blue tooth inbuilt android mobile wireless sensor network as well.

Reference

1] A. El-Sawah, N. Georganas, and E. Petriu, "A prototype for 3-D handtracking and gesture estimation," IEEE Trans. Instrum. Meas., vol. 57,no. 8, pp. 1627– 1636, Aug. 2008.

[2] D. G. Lowe, "Distinctive image features from scaleinvariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, Nov. 2004

[3] A. Bosch, X. Munoz, and R. Marti, "Which is the best way to organize/ classify images by content?" Image Vis. Comput., vol. 25, no. 6, pp. 778–791, Jun. 2007.

[4] H. Zhou and T. Huang, "Tracking articulated hand motion with Eigen dynamics analysis," in Proc. Int. Conf. Comput. Vis., 2003, vol. 2, pp. 1102–1109.

[5] B. Stenger, "Template based hand pose recognition using multiple cues," in Proc. 7th ACCV, 2006, pp. 551–560.

[6] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multiscale color features, hieracrchichal models and particle filtering," in Proc. Int. Conf. Autom. Face Gesture Recog., Washington, DC, May 2002.

[7] A. Argyros and M. Lourakis, "Vision-based interpretation of hand gestures for remote control of a computer mouse," in Proc. Workshop Comput.Human Interact., 2006, pp. 40–51.

[8] Wu X, Su M, Wang P. A hand-gesture-based control interface for a car-robot. IEEE/RSJ International Conference on Intelligent Robots and Systems. Taipei, Taiwan; 2010. p. 4644–8.

[9] Mansmann F, Vinnik S. Interactive exploration of data traffic with hierarchical network maps. IEEE Trans Visual Comput Graph. 2006; 12(6):1440–9.

[10] Ahn H-S, Chen YQ, Moore KL. Iterative learning control: brief survey and categorization. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews. 2007; 37(6):1099–121.

[11] Slaughter DC, Giles DK, Downey D. Autonomous robotic weed control systems. Science Direct, Computers and Electronics in Agriculture. 2008;

[12] Han YM. A low-cost visual motion data glove as an input device to interpret human hand gestures. IEEE Transactions on Consumer Electronics. 2010 May.

BIOGRAPHIES

ch.swathi currently a PG scholar of Embedded Systems in ECE Department. She received B.TECH degree from JNTU. Her current research interest includes Analysis & Design of Embedded System.