International Journal of Recent Engineering Science (IJRES),
ISSN: 2349-7157, *volume1 issue6 November to December 2014*

Wireless Networked Embedded System with Real Time Kernel

R.Jegadees, Dr.U.Yaluni
Research scholar, Assistant Professor,
Department of Computer Science and Engineering,
KSK College of Engineering and Technology, India

Abstract

Emerging low power, embedded wireless sensor devices are useful for a wide range of applications, yet have very limited processing, storage and energy resources. As the range of application grows, the need arises to network several embedded systems to perform complex tasks. The use of small, cheap and customizable embedded sensors capable of wireless communication among each other provides the possibility to develop new versatile systems, with very high new functionalities. In this paper, we describe the development process of small low cost and powerful wireless embedded systems based on open source kernel and ZigBee protocol.

Key Terms: WSN, ZigBee, RTOS, Kr51, EDF.

1. Introduction

As wireless-sensor technology improves, an increasing number of organizations are using it for a wide range of purposes. Wireless embedded systems consist of number of sensors, actuators, controllers and data intensive devices. The wireless embedded systems are used in many applications such as industrial automation, Airport control systems, and home automation.

A networked sensor application includes a collection of sensors that continuously monitor the surrounding environment and collection of processors that combine all the sensory data and process according to these data. Embedded processors with kilobytes of memory are used to implement complex, distributed, ad-hoc networking protocols.

Wireless sensor networks are distributed computing systems that have many limitations. These include limited computing power, low memory and have low bandwidth available for communication. Sensor networks are very data-centric, meaning that the information that they collected from their environment must be delivered in a timely fashion to the destination.

The important issues in the development process of wireless embedded system are real-time constraints (messages arrive with bounded delays), resource constraints (power, bandwidth), unreliability of sensor nodes, and sensor density for the area of coverage [8]. As large-scale real time sensor networks are developed, an architecture including co-operating layers of the communication protocol stack will be needed to guarantee real time requirements and reduce design and implementation cost.

The performance of the wireless embedded system is measured in terms of meeting the accuracy and delay requirements of the observer. Other performance metrics may include the life span of sensor nodes in the network, cost effectiveness, ease of deployment, reliability and scalability.

The rest of the paper is summarized as follows: next section gives the general architecture of wireless embedded node. Wireless communication protocols for wireless sensor network, routing in WSN and need for synchronization between nodes are discussed in section3. In Section 4 real time protocols and scheduling communication with ZigBee is discussed. Section 5 gives some kernel concepts and section 6 gives the implementation of wireless protocol for an application.

2. System Architecture

The general design provides an open platform for experimenting with wireless embedded networks and MICA implementation of architecture is shown in fig.1. It must be compact, low power and flexible in order to meet a wide range of experimental goals.

It has a central micro-controller that performs all sensing, communication and computation. A real time OS is used to handle multiple tasks defined in a Microcontroller concurrently. RF transceiver is used for communication. Power management subsystem is designed to regulate the supply voltage of the system. I/O subsystem interface is an expansion connector designed to interface with a variety of sensing and programming boards. The expansion connector can also be used to communicate with other devices such as PC. External flash memory is used to provide persistent data storage.

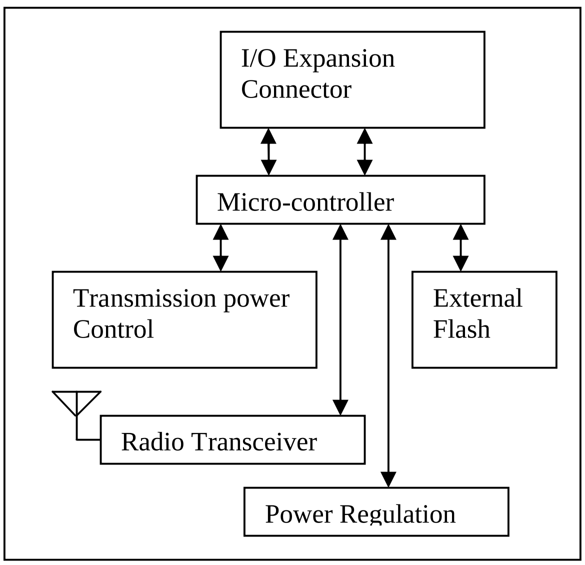

Fig 1. Mica Node Overview

3.Wireless Communication Protocol

Any communication protocol for sensor networks must ensure that messages are delivered with bounded delay to meet real-time deadlines. For the last few years, we have used a great expansion of remote control devices in our day-to-day life. As the range of application grows, the number of devices to control will increase. To interact with all these remotely controlled devices, we will need to put them under a single standardized control interface that can interconnect into a network. Many wireless protocols are used in various applications and the various protocols are summarized in table1.

Table1. Wireless Protocols

Protocol	ZigBee 802.15.4	Bluetooth 802.15.1	WiFi 802.11b
Applications:	Monitoring and control	Cable replacement	Web, Video, email
Data capacity (Kbps):	250	1,000	11,000+
Range (meters):	70	10	100
Battery life	Years	days	hours
Nodes per network	255-65000	8	30
Software size (Kbytes):	4 - 32	250	1,000+

The main requirements for devices in such types of networks are :

1. Extremely low power consumption
2. The ability to sleep for a long time
3. Simplicity
4. Low cost

One of the most promising wireless protocols is ZigBee, a software layer based on the IEEE 802.15.4 standard. The main features of this standard are network flexibility, low cost, very low power consumption, and low data rate in an ad-hoc self-organizing network among inexpensive fixed, portable and moving devices[9]. The data rate is 250kbps at 2.4GHz, 40kbps at 915MHz and 20kbps at 868MHz. The operational range of ZigBee is 10-75m.

IEEE 802.15.4 is now detailing the specification of PHY and MAC by offering building blocks for different types of networking known as star, mesh, and cluster tree. Network routing schemes are designed to ensure power conservation, and low latency through guaranteed time slots. A unique feature of ZigBee network layer is communication redundancy eliminating single point of failure in mesh networks. Key features of physical layer include energy and link quality detection, clear channel assessment for improved coexistence with other wireless networks.

3.1. Routing Protocol

In general two styles of routing are used in any network

1. Distance rector routing
2. Link state routing

In distance rector each node announces its routing table to its neighbors whereas in link state routing it floods its routing table. Both update their routing tables periodically.

ZigBee routing protocol is based on the Distance vector routing, in which each ZigBee router which transmits the frames from a particular source to a particular destination with the help of routing table. Routing table consists of an entry for each route and this entry has the details of logical distance to the destination and the address of the next router.

In ZigBee, routes are established on-demand using a route discovery process is shown in fig.2 and in this process two originating device broadcasts a route request command and the destination device sends back a route reply[6].

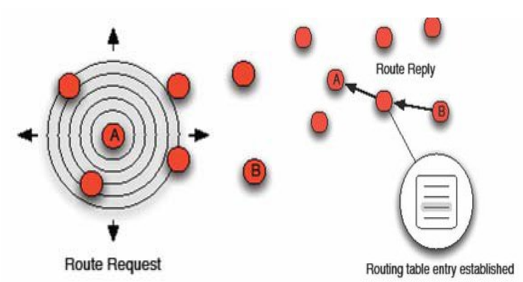

Fig.2 Path Discovery Process in ZigBee

When a source node needs to communicate with another node for which it has no routing information in its table, the Path Discovery process is initiated. The source node initiates path discovery by broadcasting a route request (RREQ) packet to its neighbors. As the RREQ travels from a source to destinations, it automatically sets up the reverse path from all nodes back to the source. When RREQ reaches the destination it sends a route reply packet (RREP) to the source node through the reverse path. The source node can begin data transmission as soon as the first RREP is received, and can later update its routing information if it learns of a better route.

3.2. Synchronization Protocol

Time synchronization is essential for most network applications. ZigBee has very low power consumption requirements. 802.15.4 network can operate in either master-slave or peer-to-peer configuration. ZigBee can accommodate master-slave networks that can be more power-efficient. The master-slave would be common in many sensor-based applications where a number of very limited slaves are controlled by a master controller. ZigBee supports both mesh network and a star network topology in which a node arts as master for a number of other slave nodes. The master sends periodic beacon messages to the slaves, providing regular windows of time for the nodes to communicate.

Synchronized clocks are essential for shared channel communication protocols. For time synchronization to work, there is a fixed point in time is needed[4], with the timestamp in the given message. In ZigBee[2], end of the Start of Frame Delimiter (SFD) is used to give that fixed point.

The sender makes a time stamp immediately, after it has transmitted the SFD and inserts the timestamp into the message. The receiver makes a timestamp, when it receives the SFD and stores it with the message.

4. Real time protocol

In wireless sensor network, messages in the network are delivered according to their real-time deadlines. Some mechanism is required to assign priorities to packets that are to be sent on the sensor network. Three different strategies are presented [7], first is a MAC layer solution and is based on the IEEE 802.11 standard. In this solution, sensor nodes are arranged in a cellular structure and perform Earliest Deadline First (EDF) analysis to access wireless medium. The second solution is a network layer solution and is designed to guarantee that messages move across the network (from source to destination) at a predetermined speed. Final solution provides real time packet prioritization and scheduling.

In [3] two fundamental aspects that constrain the real-time behavior are the medium access control (MAC) protocol and the mechanisms to handle dynamic communication requirements. In ZigBee, the MAC coordinates transceiver access to the shared radio link. It also schedules and routes data frames. This provides address generation and address recognition, and it verifies frame check sequences. To schedule the frame transmissions in Beacon-less mode, 802.15.4 uses a form of carrier sense multiple access collision avoidance (CSMA-CA). Each device listens before transmitting to decrease the likelihood of two transmissions occurring at once. If a transmitting node suspects that it has clashed with another transmission, it will roll back and reschedule itself to transmit later.

Scheduling communication in real time applications

Scheduling is an important service provided by RTK to handle multiple tasks. Scheduler is designed to schedule all messages that are ready for transmission. When invoked, the scheduler manages the current schedule and all the newly arrived messages at each node. We consider the following system constraints in [5],[7] to design the scheduler for real time applications:

1. Network Interface Constraint: At any instant, a node can be either a sender or a receiver, but not both.
2. Range Constraint: A node can receive a message only, if it is within the sender's range.
3. Interference Constraint: Two simultaneous transmissions will not interfere if and only if both receivers are mutually outside the other sender's range.

Many scheduling approaches are used to satisfy real time constraints in real time applications. The scheduling algorithms are: co-operative scheduling, Pre-emptive scheduling and Earliest Deadline First(EDF) scheduling. Co-operative scheduling is based on fixed time-slice and is working with FIFO basis.In pre-emptive scheduling at any time the highest priority task can interrupt the system and is based on fixed priority, fixed time slice and critical section. EDF is a dynamic scheduling algorithm and is based on time deadline.

4. About RTK

Now-a-days, RTOS (Real Time Operating System) takes more and more important role in industry. An RTOS becomes essential when there is processing and servicing of multiple devices. RTOS is an operating system with certain real time resource schedule and communication capability. RTOS may be event driven RTOS & time sharing RTOS. Event driven RTOS is used for the system that changes state only in response to an incoming event and time sharing RTOS is used for the system that changes state as a function of time.

The heart of the RTOS is the kernel. A Kernel is the central core of an operating system and it takes case of all the OS jobs like booting, task scheduling and standard function libraries etc. and also RTK (Real Time Kernel) provides an abstraction layer that hides the hardware details from the application software.

Kernel provides the services of task management, inter-task synchronization and communication, timer services, dynamic memory allocation and device management to application software.

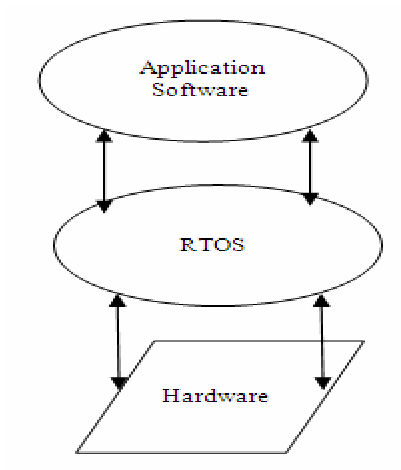

Fig. 2 Kernel Function

Multiple tasks are handled by RTK. There are some common variables (or) functions that are used in many tasks. These common variables (or) functions are declared as critical section to avoid repeated modification. There occurs a problem, when two tasks access the critical section data concurrently. To avoid this problem, semaphore will be used. Semaphore provides a mechanism to let a task wait till the other one finishes. It is a way of synchronizing concurrent processing operations.

6. Implementation

This section shows the implementation of the application based on [1]. The system minimizes the energy and cost by getting the power from the environmental sources such as wind and solar. The system is formed with three nodes, one is considered as master and the other two are slaves. The load(any electrical device) is connected to the master. The system provides power according to the demand and if the demand exceeds the master shut down the electrical devices. The other two client nodes monitor the environmental condition using sensors and send that data to server node. According to the sensory data from clients server issues control commands to the network.

The master node consists of sensor connected with micro-controller. The sensor output is digitized and converted to a form possible for transmission. The data from the micro controller is then sent to the ZigBee transceiver. So, each node contains a Wireless protocol ZigBee which interface to a host device through a logic-level asynchronous serial port. The ZigBee module is configured for support Broadcast Mode, in which any RF module will accept a packet that contains a broadcast address.

RTK- Kr51

The KR-51 executive is a real time kernel for the 8051 microcontroller family. Placed in the heart of an application, it provides efficient task scheduling by driving task activation and task relationship. Kr51 is designed to support applications with real-time requirements and provide features such as full pre-emptability, minimal interrupt latencies, scheduling policies and interrupt handling mechanisms.

In the design of kernel following functions are used:

```
void kr_create (unsigned char task_id)
void kr_delete (unsigned char task_id)
void kr_del_abs (unsigned char time)
void kr_sendsig (unsigned char task_id)
char kr_tst (char time-out, char sem_num)
void kr_free (char sem_num)
```

Kr51 provides three types of services are system services, information services, utility services.KR51 system services provide the interface between user tasks and the kernel. Information services provide functions that return information relative to the kernel and utility services provide some special functionalities like mailbox, semaphores etc. KR51 requires the LX51 linker and RC51 C compiler programs, and the RIDE Integrated Development Environment is used to simplify project management.

Conclusion

In this paper, we have presented the issues involved in wireless embedded system with multiple sensor nodes. Sensor nodes have limited resources including computation power, communication bandwidth, and battery power. This paper also describes some real time and wireless protocols used for real time applications to meet real time deadline. ZigBee supports the MAC layer protocol and the system is implemented with ZigBee. The real time kernel is designed with embedded node to meet real time constraints in the network.

References

1. Handa T., Ichimura, J., Nishi H., Roesener, C. (2007). KNIVES: Network based Demand and Supply Control System - Enhancement for Network and Environmental Affinity. Industrial Informatics, 5th IEEE International Conference on Volume 2, Issue 23-27, June 2007 Page(s):1171 – 1176.
2. Cox, D., Jovanov, E., Milenkovic, A. (2005) Time Synchronization for ZigBee networks. In: IEEE System Theory Symposium. pp135-138
3. Facchinetti, Tullio, Giorgio Buttazzo and Luis Almeida (2005). Dynamic reource reservation and connectivity tracking to support realtime communication among mobile units. In: EURASIP Journal on Wireless Communications and Networking.
4. Hui Dai, Richard Han (2004). Tsync: a lightweight bidirectional time synchronization service for wireless sensor networks. In: Mobile Computing and Communications Review. pp. 125–139.

5. Li, Huan, Prashant J. Shenoy and Krithi Ramamritham (2004). Scheduling communication in real-time sensor applications. In: IEEE Real-Time and Embedded Technology and Applications Symposium. pp.10-18.
6. Scholten, Hans, Pierre G. Jansen and Laurens Hop (2004). Routing in wireless multimedia home networks.. In: Proceedings of the IEEE International Conference on Multimedia and Expo on Volume 3, June 2004. pp. 1799-1802.
7. Stankovic, John A., Tarek F. Abdelzaher, Chenyang Lu, Lui Sha and Jennifer C. Hou (2003). Real-time communication and coordination in embedded sensor networks. Proceedings of the IEEE 91(7), 1002–1022.
8. Buttazzo, G. C. (1997). Hard real-time computing systems-predictable scheduling algorithms and applications. In: Kluwer Academic Publishers.
9. http://ZigBeealliance.com

.