POWER OPTIMIZATION FOR GEOGRAPHIC ROUTING ALGORITHM IN WIRELESS SENSOR NETWORK USING TI MSP430

By

MITUL PATEL

07MCE015

INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NIRMA UNIVERSITY OF SCIENCE & TECHNOLOGY

AHMEDABAD-382481

MAY 2009

Power Optimization For Geographic Routing Algorithm in Wireless Sensor Network using TI MSP430

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Mitul Patel

07MCE015

INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NIRMA UNIVERSITY OF SCIENCE & TECHNOLOGY

AHMEDABAD-382481

May 2009

Certificate

This is to certify that the Major Project entitled "Power Optimization For Geographic Routing Algorithm in Wireless Sensor Network using TI MSP430" submitted by Mitul Patel (07MCE015), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. S.N. Pradhan
Guide and Professor,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Prof. D. J. Patel
Professor and Head,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Abstract

A wireless sensor Network consists of a small battery operated sensors. After successful deployment of sensor nodes, it is often unfeasible to re-charge sensor nodes or replace batteries. Another major problem in wireless sensor network is the broken connectivity between sensor nodes due to power sources that drain out of power. To overcome these major drawbacks, optimized power consumption strategies need to be developed either by doing optimization in hardware or by using efficient software techniques.

Power consumption constraints in WSNs require efficient task distribution among various components of wireless sensor node (i.e. radio transceiver, sensing device and computation device). Most techniques use method that will switch off transmitter & receiver according to some strategy to save power but they do not consider power saving in the processor. This project proposes novel approach for routing algorithm of wireless sensor network that will optimize the power using ultra low power micro controller. This ultra low power micro controller works in different low power modes which have different power consumption statistics. The different routing algorithms have different requirements for execution on various sensor node components so based on requirements, the processor need to schedule the code so that processor will use best low power mode to execute that code on different nodes.

In real application like routing in sensor network, most of the time sensor node has only one or few components that are in active state. So the power of the remaining components that are inactive will simply be wasted. The proposed routing method will put the processor into low power mode whenever it finds the components in the inactive state. When component is once again in the active state, that component will generate an interrupt which will change the mode of the processor from low power mode to regular mode.

Acknowledgements

The successful completion of a project is generally not an individual effort. It is an outcome of the cumulative efforts of a number of persons, each having own importance to the objective. This session is a vote of thanks and gratitude towards all those persons who have directly or indirectly contributed in their own specials way towards the completion of this project.

It gives me great pleasure in expressing thanks and profound gratitude to my project guide Dr. S. N. Pradhan, M.Tech (CSE) Program Coordinator, Department of Computer Science & Engineering , Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout the project.

I would like to give my special thanks to Prof. D. J. Patel, Head, Department of Computer Engineering, Institute of Technology, Nirma University for allowing to do project work under the university campus.

I extend my sincere thanks to my family members and my friends who have always been my source of inspiration and motivation.

- Mitul Patel

07MCE015

Contents

List of Tables

List of Figures

Abbreviations

ACLK	Auxiliary Clock
ADC	Analog-to-Digital Converter
CPU	Central Processing Unit
CISC	Complex Instruction Set Computing
DCO	Digitally Controlled Oscillator
GIE	General Interrupt Enable
GUI	Graphical User Interface
ISR	Interrupt Service Routine
LPM	Low Power Mode
MAB	Memory Address Bus
MCU	Micro Controller Unit
MCLK	Master Clock
MDB	Memory Data Bus
NMI	Non Maskable Interrupt
PC	Program Counter
RISC	Reduced Instruction Set Computing
SCG	System Clock Generator
SFR	Special Function Register
SMCLK	Sub-system Clock Generator
SR	Status Register
WDT	Watchdog Timer

Chapter 1

Introduction

1.1 General

Wireless sensor network(WSN) technology is promising and is therefore gaining popularity day by day in a wide area of different applications. The WSN nodes operates on battery power which is often deployed in a rough physical environment; changing the batteries is therefore a complicated task, as some networks may consists of hundreds to thousands of nodes. Such large physically distributed networks increase the difficulty of changing batteries and makes recharging almost impossible during operations. This problem has forced node, network and system developers to make changes in the basic WSN architecture to minimize the energy consumption especially of the nodes in order make the network and overall system application more energy efficient. The sensor node architecture is given in 1.1.

As shown in 1.1, there are main three components: Sensing unit, Processing unit and Transceiver unit. So the first step in order to reduce the power consumption is to decide which component is consuming how much power. The main task of the processing unit is to control the sensing parts and to do computation on data coming from sensor and produce output. It is also associated with executing of

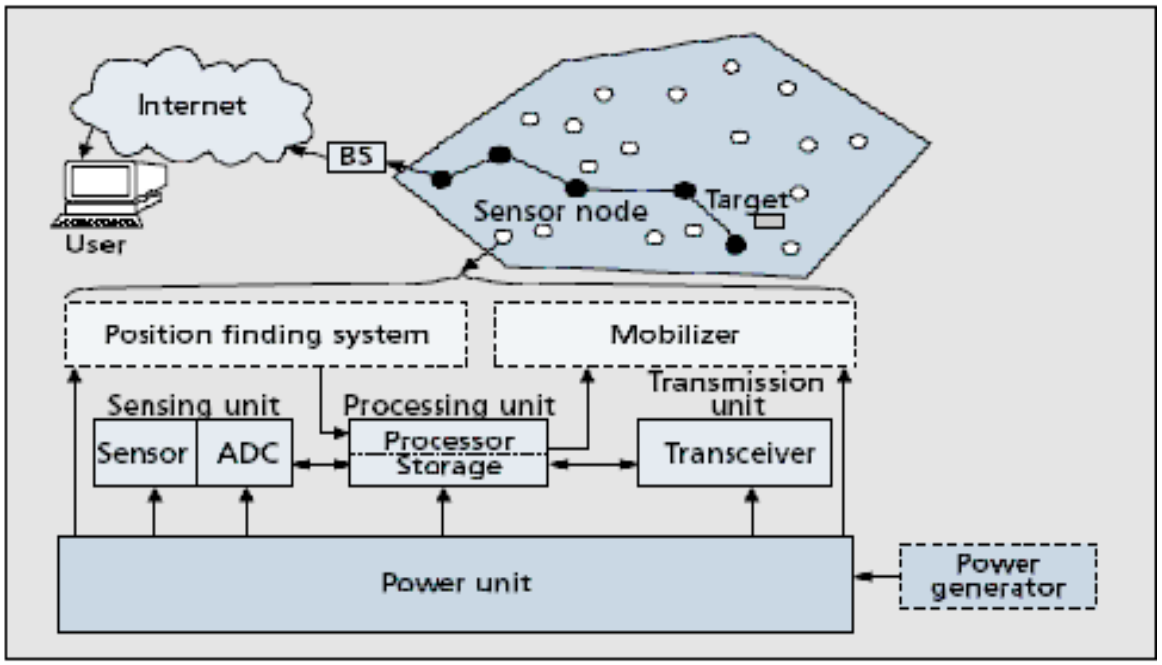


Figure 1.1: Sensor Node Architecture

communication protocol. Radio transceiver consumes maximum power in idle state because it has mainly three states in which it is working: active, idle and sleep. In idle state it will consume as much power as in reception state. So whenever there is no need for radio transmission then completely switch of the radio instead of putting it in an idle state.

1.2 Motivation

Wireless sensor network is a network of typically small battery powered wireless devices. Once deployed, it is often infeasible or undesirable to re-charge sensor nodes or replace their batteries. Thus, energy conservation becomes crucial for sustaining a sufficiently long network lifetime. Due to the nature of wireless communication, one performance metric of the network can be affected by various factors across layers. So there is a need to work upon different layer simultaneously that will help in reducing the power consumption and increasing the network lifetime.

The second major issue in wireless sensor network is the connectivity between various sensor nodes. As the connectivity is needed to transfer information from various sensor nodes to the sink node, so to increase the network lifetime nodes has to do energy efficient computation and communication in order to reduce the power consumption of individual node. As in wireless sensor network, each end device will generally communicate with another end device through access point in a centralized architecture so the power consumption of an access point is more. Therefore connectivity between the access point and number of end devices is a crucial task to do.

1.3 Organization of work

The power consumption of wireless sensor node can be optimized either by using energy efficient algorithms or by using special hardware devices which have ultra low power consumption characteristics. Rather then focusing on addressing the problems related with increasing the amount of available power this thesis focuses on design techniques for minimizing the amount of power required by the wireless sensor system.

The goal of this thesis is to design novel approach for routing algorithm that will efficiently use the low power and power switching characteristics of micro controller. As there are many choices available for processors like PIC micro controller, AVR micro controller and MSP430 micro controller from Texas Instruments. Among these micro controllers, Texas Instrument's MSP430 micro controller has the ultra low power consumption in terms of miliwatt(mW)

In this project, architecture of MSP430 micro controller is discussed and try to understand power consumption characteristics by using simple communication protocol that will use to transfer information between sensor nodes. The main focus of this project is to design novel approach for routing algorithm used in wireless sensor network that will use this micro controller. The tool that is to be used to develop and test the power consumption of this routing algorithm is the virtual lab provided

by Texas Instruments.

1.4 Thesis Report Organization

Chapter 2, *Literature Survey*,In it discussion about different techniques which includes various MAC layer protocol and routing techniques that has been suggested in order to optimize power by various researchers is provided.

Chapter 3, *MSP430 micro controller & development tools*, presents explains general architecture of MSP430 Ultra Low Power Micro controller with some detail information about various low power operating modes and instruction set. This chapter also describes detail description of various development tools for MSP430 micro controller.

Chapter 4, *Position Based Routing*, detail description of position based routing taxonomy is discussed. It also includes discussion of geographic routing algorithm.

Chapter 5, *Implementation, Results and Analysis*, It describes new routing method and also describes a new way of using micro processor various low power modes. The simulation results along with the performance analysis of the proposed algorithm are presented.

Chapter 6 concluding remarks and scope for future work is presented.

Chapter 2

Literature Survey

As energy saving of wireless sensor network is one of the hot topics in the Wireless Sensor Network field much research has already been done and more is expected. Much of the recent works has targeted a single factor influencing energy consumption of a network either at hardware or communication level. This project however has been done by identifying and collecting the most influential of the factors effecting energy efficiency in Wireless Sensor Network.

2.1 Sources of power consumption

The sources of power consumption, with regard to network operations, can be classified into two types: communication-related and computation-related.

Communication involves usage of the transceiver at the source, intermediate (in the case of ad hoc networks), and destination nodes. The transmitter is used for sending control, route request, and response messages, as well as data packets originating at or routed through the transmitting node. The receiver is used to receive data and control packets, some of which are destined for the receiving node and some of which are forwarded.

Understanding the power characteristics of the mobile radio used in wireless devices is important for the efficient design of communication protocols. A typical

mobile radio may exist in three modes: transmit, receive, and standby. Maximum power is consumed in the transmit mode, and the least in the standby mode. Thus, the goal of protocol development for environments with limited power resources is to optimize the transceiver usage for a given communication task.[1]

2.2 Energy Saving At Node Level

The energy saving methods can be affected or can be classified under two heads:

Device Level Hardware component selection and their configuration to achieve low energy consumption in a wireless sensor node.

Network Level Choice of communication methods and protocols to minimize energy consumption.

2.2.1 Overall design of sensor node

In a Sensor node there are four essential parts: processing unit, sensing unit, transceiver unit and, power unit. This part of Wireless sensor mote (WSM) is built on the Integrated Chip (IC). One needs to choose proper Peripheral of WSM and configure the entire network which will be more energy efficient. The basic diagram of wireless sensor mote is shown in 1.1.

Processing unit is a part of micro controller unit which can read sensor data, perform some minimal computations and make a packet ready for transfer in the wireless communication channel. The local memory requirements will not be high and emphasis will be placed on the modes of operation to facilitate low-power operation. The Communication module/unit is typically an RF transceiver that should support the 802.15.4. This unit helps in collecting information and to exchange or control data acquisition. The maximum amount of energy is used in communication module when compared to the two other modules. Sharing information between sensor nodes will consume more amount of energy than implementing the calculation within individual

node. In Sensing unit, sensors are literally used for sensing temp, images, gas etc. Sensors to sense different things require different amount of energy. In Power unit, base stations are more often connected to main power supply, whereas nodes in the network depend on batteries to supply power. Hence there is a requirement to choose power efficient hardware and various efficient-operation modes to make the network more power efficient[2]. Figure 2.1 reflects power consumption of WSN in various states.

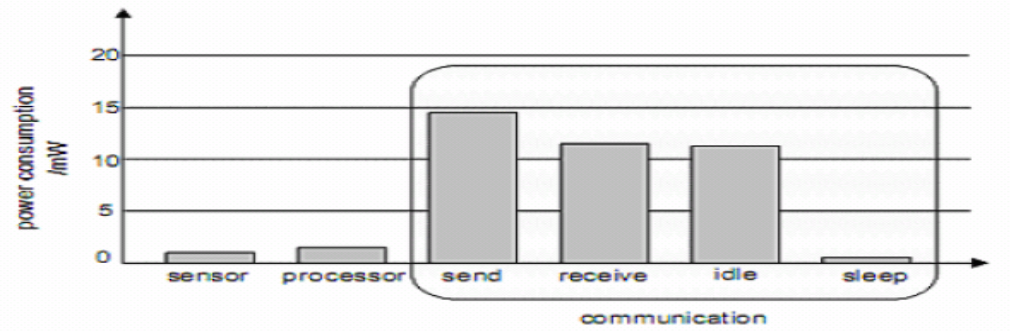


Figure 2.1: Power Consumption of Sensor [2]

2.3 Energy Efficient MAC protocols

Medium Access Control (MAC) synchronizes the channel access in an environment where numerous nodes access a shared communication medium. The MAC Layer provides an interface between the data-link layer and the PHY layer. Furthermore it can handle data service and MAC management service like generating networks beacons if the device is a coordinator, as well as GTS allocation and synchronizing to the beacons which is important for data polling and energy saving purposes. The MAC layer has a major role in making the network more power- efficient. Basically MAC is responsible for the coordination between neighbors. MAC protocol must not only be an energy constrain protocol but should also be able to address scalability

issues (handling dynamic changing topology of WSNs, unaffected by node density etc) and take into account the timing constraints of the applications. A large amount of energy can be saved at this layer. Some of the causes behind energy wastage in wireless sensor networks related to this layer are mention in the short description below.

Collision Once a collision of packets occurs there is re-transmission of the collided packets that were discarded, resulting in wastage of energy due to re-transmission. This can be saved by avoiding collisions in the first place resulting in conserving network energy.

Idle listening In contention-based MAC protocols when nodes are neither transmitting nor receiving data, channel still tries to sense data in which results in the wastage of energy.

Control packet overhead The number of control-packet should be reduced to save energy since the transmission, reception and listening of these packets will result in energy consumption.

Overhearing When node receives a packet which is addressed to some other node, overhearing occurs. This takes place during high traffic loads. Overhearing unwanted packets is wastage of energy.

All these factors must be taken into account by an energy aware MAC protocol in order to minimize the energy wastage in WSN.

2.3.1 Contention Free Protocols

Many nodes are present in sensor network, and they are most likely distributed in non - uniform direction. When two sensor nodes attempt to access the communication channel at the same time, contention occurs. Due to contention messages could collide

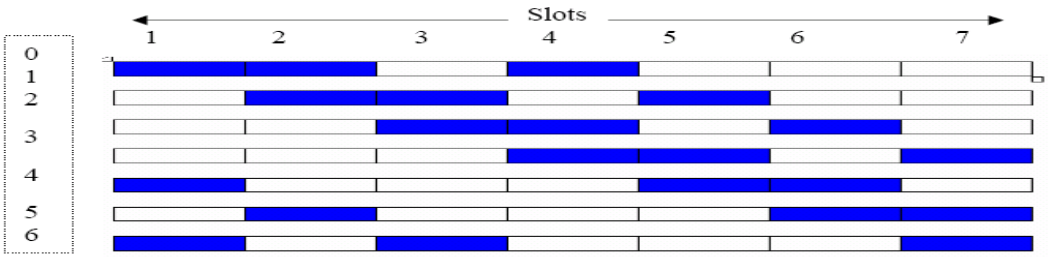


Figure 2.2: An example of slot based protocol [2]

when traffic is frequent. As a result collision has an effect on the network life time of a sensor network A MAC protocol is believed to be contention-free if it does not allow collision. All accessible contention free MAC protocol assumes that the sensor nodes are time-synchronized in some way. In contention-free protocols, the channel is most often divided into time slots. To send the data, each node uses the time slot and thus it provides collision free communication.

Slot-based protocols

The main idea of this protocol [2] is that the time is divided into periods which has a certain number of fixed slots, where some active slot are assigned to keep the nodes active, send the beacons to neighbors and to listen for message acknowledgement or listen to requests from neighbor nodes. In slot-based protocol, beacon helps to communicate with any two nodes of a network. Figure 2.2 demonstrate that time is divided in seven periods. Any two neighbors can eventually hear each other if they use an 1101000 activation schedule since they always have at least one overlapping slot.

Schedule of the form 1101000 where 1 stand for active stage and 0 for inactive. Nodes are first communicated with each other through beacons and wait for one of them propel data to neighbors. There are no pre-defined activation schedules for slots. But it has been already proved that the number of active slots is k, then k

= μ+1, where k stand for active slots, M for number of overlapping and μ is prime number, as a result if . M = 1(minimum overlapping) and number of slots in the period is t then active schedule exits for t = $\mu^2 + \mu + 1$, if number of overlapping is increased then latency will be less, where as energy consumption will be high.

Time Division Multiple Access (TDMA) Protocol

TDMA is a channel access method that is used to share a radio link. In this technology, different users can share the same frequency channel by dividing the signal into different time slots. Idle listening can be avoided by using TDMA protocol and MAC layer contention issue can be fixed by scheduling transmissions earlier, so that the nodes can have information in advance when the radio should be turned on and doing so they can stay away from collision. Figure 2.3 shows time slot scheme in TDMA protocol where time is organized as sections of signalling slots and transmission slots. In classical TDMA protocols all the nodes can see each other while master node initiates the super frame in specific time interval for network operation. This technology is most suitable for single hop but for multi hop it is very difficult for multiple simultaneous transmissions.

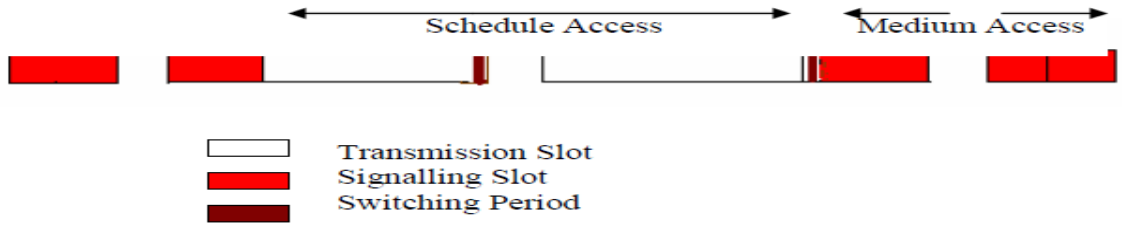


Figure 2.3: Time slot organization [2]

One of the energy efficiency TDMA protocol is Traffic-Adaptive Medium Access (TRAMA) [1] that was developed based on Node Activation Multiple Access (NAMA) for WSN. In that proposal, time is divided into time slot and each nodes

channel access is organized over a fraction of time. In this mechanism each node has information of other two hop neighborhood and transmits data freely without interferences according to their winning slots, which mean the neighbors who have highest precedence according to their slot number and node ID. TRAMA assume low data rate compared to the time slot. Random access periods are used for time synchronization to exchange the information of the neighbors. First hop neighborhood information is broadcast in contention-based slots but the actual data transfer takes place according to a contention-free schedule.

To make this protocol energy efficient, TRAMA change nodes to sleep state whenever possible and attempt to reuse transmission slot which are not use in transmission. It is quite possible when one selected node allow his transmission slot to be used by another node if it does not have any packet to send. To make use of low power, idle mode and reuse the transmission slot nodes can share existing traffic with their neighbors.

2.3.2 Contention Based Protocols

In contention-based protocols, a given transmit chance towards a receiver node can in principle be taken by any of its neighbors. If two or more neighbors try their luck at the same time, they have to compete with each other. In this case a collision might occur, wasting energy for both transmitter and receiver. In Contention-based protocol, stress is made on minimizing collisions rather than avoiding it completely (Schedule-based or collision free protocols). In this technology all node shared single radio channel according to their demand. To reduce the probability of collision or to avoid collision distributed algorithm is used which allocates the channel between nodes. In this mechanism nodes listen to the channel before sending a message just to ensure that channel status is free before transmission. If all channel are busy then node continuously listen until the medium is free to send data.

However a CSMA-based protocol in multi-hop wireless networks guides towards

hidden node problem. Due to insufficient radio coverage such as in figure 2.4, node n1 is not able to talk directly to node n2, n1 then tries to send data to its immediate neighbor R but node n2 will be unaware about this transmission and in the mean time n2 can also initiate to send data to R. As a result collision will occur on the receiving node R.

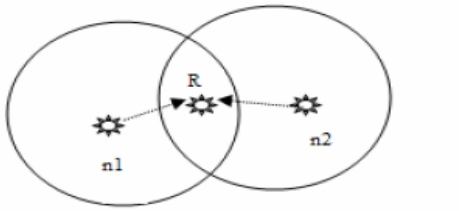


Figure 2.4: Hidden Node Problem [2]

To handle this problem additional signalling control messages have been proposed. One of them is out of bound signalling which relies on sending busy tone when transmission is in progress hence preventing other nodes from initiating transmissions. This system is known as Busy Tone Multiple Access which eradicates the hidden terminal problem. Busy tone radio is cheaper than other methods and consumes less power.

Sensor MAC(S-MAC)

S-MAC supports multi-hop operation. Its key features are:

- Periodic listen and sleep
- Collision avoidance
- Overhearing avoidance
- Fixed duty cycle

Sensor MAC (S-MAC) uses three new procedures to decrease energy consumption and support self-configuration. It is a contention-based protocol with low duty cycle. For SMAC, energy consumption in idle listening is to be reduced by allowing neighbouring nodes of transceiver and receiver to sleep periodically during transmission, by doing so this scheme put nodes into low duty cycle. Figure 2.5 reflects SMAC listen sleep schedule. S-MAC is based on contention. Periodically sleeping is good in low traffic cases. If a node can sleep for longer time it consumes less energy. For example, if the duty cycle is trimmed down to 50can sleep half a second and be active for the other half, which results in 50

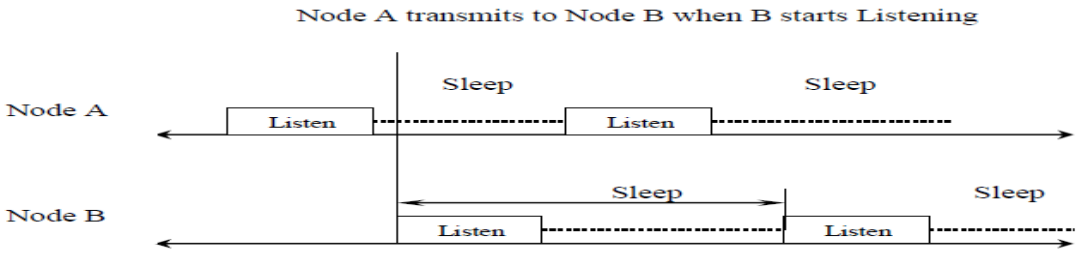


Figure 2.5: S-MAC listen/Sleep Module [2]

The S-MAC approach is to split a long message into small parts and send them out in bursts. Nodes in the S-MAC exchange their sleeping schedule and before going to sleep nodes broadcast their schedule to their neighbours as a SYNC packet. Nodes listen to this sync message and follow it. If they do not get the sync message they make their own schedule. If the new scheduler is not used to the neighbours nodes, it can be discarded. If two nodes want to talk, the sender first will use single RTS (request to send). When it is received by another node it replies with CTS (clear to send). All the nodes will again go back to sleep mode cycle once the transmission is completed.

Time Out MAC(T-MAC)

The problem with S-MAC protocol is the idle listening time and also the fact that it has a fixed duty cycle. This makes S-MAC unsuitable for varying traffic load. When there is low traffic, a duty cycle wastes large amount of energy which are made or tuned for handling high traffic load. Similarly under high traffic conditions a duty cycle which is made for low traffic load will decrease the throughputs. Hence, an elaboration based on S-MAC, T-MAC has been developed to fix the problem by using adaptive duty cycle. If there is no movement or no event in the neighbors, the node goes to sleep. Apparently T-MAC has the same role as S-MAC under steady traffic, but it is more power-consuming in variable traffic. Comparison between S-MAC and T-MAC is given in figure 2.6.

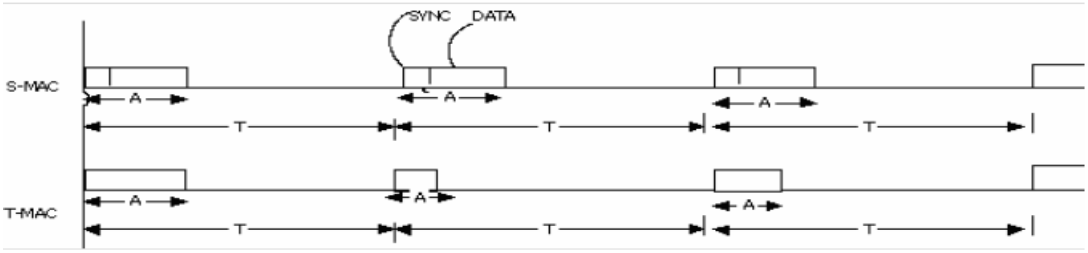


Figure 2.6: Comparison of S-MAC and T-MAC

Dynamic Sensor MAC(DS-MAC)

Dynamic sensor MAC protocol is based on SMAC. In heavy traffic load nodes can use dynamic duty cycle. When interval is too large in heavy traffic then SYNC and data message helps node to increase duty cycle and vice versa in case the traffic is low. In DSMAC, duty cycles are included in the SYNC message that they transmit. To measure the traffic condition each source calculates the queuing delay from message reception to transmission completion and inserts this to an additional field for upcoming data message. In DSMAC SYNC message updates the schedule and

then nodes updates their schedule itself and shared equally with period, which helps to decrease latency over heavy traffic. Figure 2.7 shows a DSMAC frame where the sensor node has a duty cycle twice the normal value.

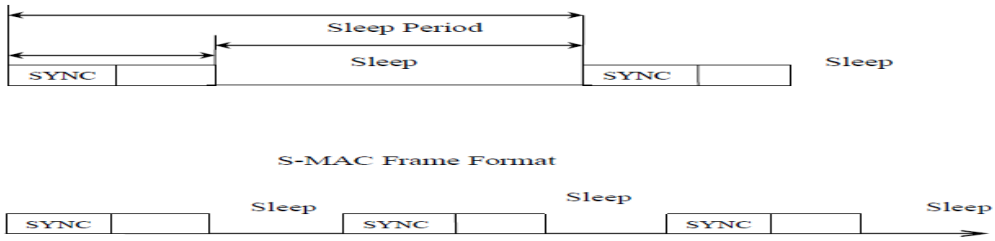


Figure 2.7: DS-MAC Frame Format

2.4 Energy Efficient Routing protocols

In minimizing the overall power consumption of the network, the data routing algorithm plays an important role. It is not necessary that the shortest route is taken for it to be the best option from an energy perspective. Information about the remaining batteries life is important and useful input for the routing algorithm. This information can be self-computed by individual nodes and broadcasted over the network for the other nodes so that the best available routing path can be chosen.

The position of cluster head and base station is also important in terms of energy efficiency, considering the fact that nodes are not placed in a uniform manner. The position of sensor nodes and data routing is pre planned in deterministic situations. Sensor nodes are placed randomly over an area of interest in most of the applications. Energy consumption has a huge impact on the setting up the routes while creating the network infrastructure. For wireless radio transmission, power is proportional to distance square or even to higher order[2].

2.4.1 Sensor Network Routing Challenges

Robustness Sensor nodes are mostly unattended and are not directly exposed to an user or end user. very often sensors are placed in a forest, hilly area or in the bed of a river or sea. Besides Sensor networks exist with the ratio of thousands of nodes per user. At such ratios, it is impossible to pay special attention to any individual node. Thus Robustness of software is most important.

Stability against task dynamics sensors may be inaccessible, either because they are embedded in physical structures, or thrown into inhospitable terrain. Thus for such system to be effective, it must provide stability.

Energy Efficiency Wireless communication, over relatively short distance, consumes significant energy. Since sensor nodes are battery operated, communication mechanism for information dissemination should be energy efficient.

Scalability The mechanism must scale to several thousands of sensor nodes in the sensor field.

2.4.2 Routing Protocols

Different energy efficient routing protocols are discussed in this section.

LEACH(Low Energy Adaptation Clustering Hierarchy)

LEACH is a clustering-based protocol that minimizes energy dissipation in sensor networks. The purpose of LEACH is to randomly select sensor nodes as cluster-heads, so the high energy dissipation in communicating with the base station is spread to all sensor nodes in the sensor network. The operation of LEACH is separated into two phases, the set-up phase and the steady phase. The duration of the steady phase is longer than the duration of the set-up phase in order to minimize the overhead. During the set-up phase, a sensor node chooses a random number between 0 and 1.

If this random number is less than the threshold T the sensor node is a cluster-head. After the cluster heads are selected, the cluster-heads advertise to all sensor nodes in the network that they are the new cluster-heads. Once the sensor nodes receive the advertisement, they determine the cluster that they want to belong based on the signal strength of the advertisement from the cluster-heads to the sensor nodes. The sensor nodes inform the appropriate cluster-heads that they will be a member of the cluster. Afterwards, the cluster-heads assign the time on which the sensor nodes can send data to the cluster-heads based on a TDMA approach.

During the steady phase, the sensor nodes can begin sensing and transmitting data to the cluster heads. The cluster-heads also aggregate data from the nodes in their cluster before sending these data to the base station. After a certain period of time spent on the steady phase, the network goes into the set-up phase again and entering into another round of selecting the cluster-heads.LEACH (Low-Energy Adaptive Clustering Hierarchy), a clustering-based protocol that utilizes randomized rotation of local cluster base stations (cluster-heads) to evenly distribute the energy load among the sensors in the network[3].

PEGASIS(Power Efficient Gathering in Sensor Information System)

It is an improvement over LEACH. It is chainbased protocol where cluster heads are chosen randomly which means that nodes will die throughout the network in a random fashion. Hence density is maintained stable throughout the network. Node communication is only possible between close neighbors. A chain is formed which starts from the cluster head, the information of network topology is assumed to be known by PEGASIS and the chain is constructed by the use of a greedy algorithm[2].

GPSR(Greedy Perimeter stateless Routing)

It is a type of Geographic Routing. Greedy Perimeter Stateless Routing (GPSR), a novel routing protocol for wireless datagram networks that uses the positions of routers and a packets destination to make packet forwarding decisions. GPSR makes

greedy forwarding decisions using only information about a routers immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region. By keeping state only about the local topology, GPSR scales better in per-router state than shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under mobilitys frequent topology changes, GPSR can use local topology information to find correct new routes quickly. Figure 2.8 summarize greedy and perimeter forwarding[3].

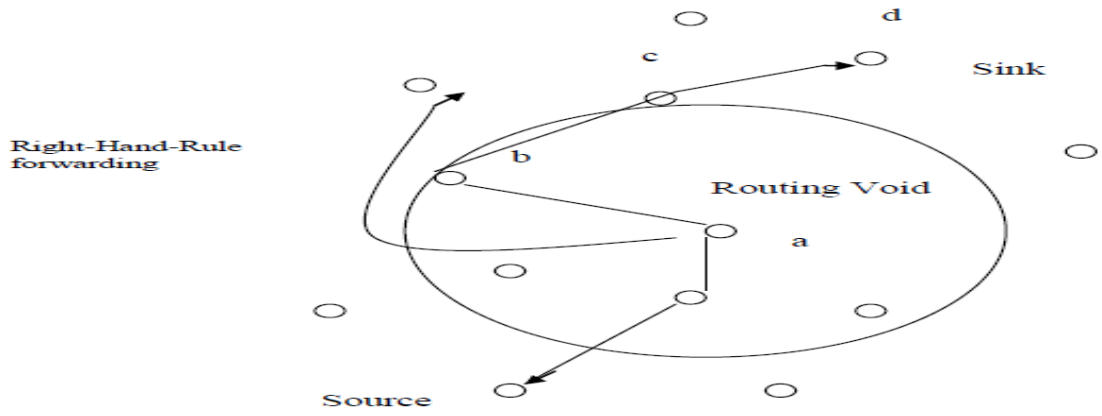


Figure 2.8: Greedy and Perimeter Forwarding[2]

GAF(Geographic Adaptive Fidelity)

In GAF, a distributed algorithm helps to find the grid leader or the backbone node which uses standard ad-hoc routing protocol AODV. Nodes alternate their states (active or sleep) and detect where they can change or take over the grid leadership. In GAF, the whole sensor field is divided in a small square grid with side L = r / $\sqrt{5}$ where r represents the radio range where backbone node (Grid leader) can talk to each other and other nodes can receive or send data to their backbone node. A Sketch of GAF square grid picture is shown in figure 2.9.

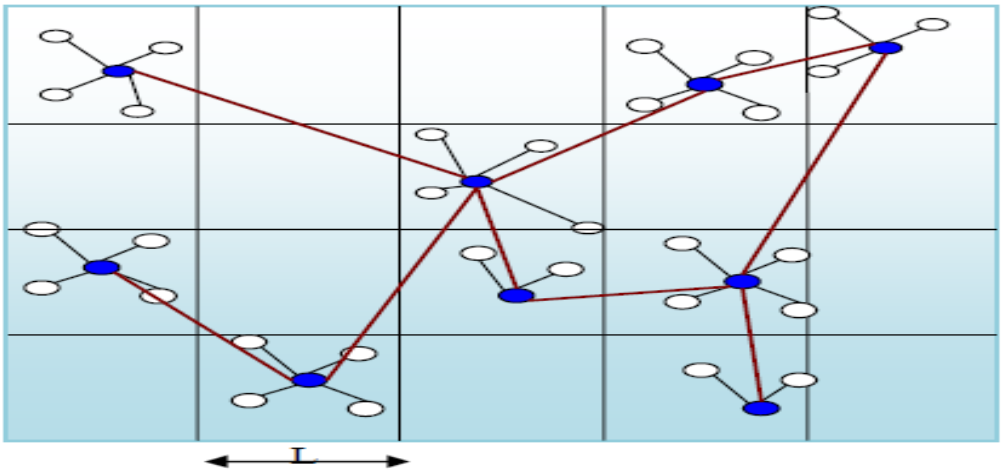


Figure 2.9: Outlines GAF grid partitioning[2]

ASCENT(Adaptive self configuring sensor network topologies)

Large sensor network consist of many nodes and in some cases all the nodes do not need to be in active mode. When few of the nodes are in use in the network, the distance between neighboring nodes may be too long which will cause the packet loss rate to increase. On the other hand data transmission energy over long distance will be scrutinized. Similarly, if all nodes of the network are in use, the system will lose unnecessary energy. ASCENT describes a logical energy efficiency and adaptive procedure for dealing with these kinds of situations. In ASCENT, nodes try to save energy for extending network lifetime by using the adaptive techniques that permits applications to configure the underlying topology based on their needs. ASCENT is a self-configuration technique reacting to operating condition measured locally. Active nodes from the network are selected by ASCENT which stays awake all the time and performs multi-hop packet routing and the rest of nodes remain inactive. These inactive nodes periodically check whether they should become active or not. These nodes are passive nodes and they remain listening to packets but they do not take part in the transmission[2].

2.5 Typical micro controller used in sensor networks

This section explores some of the most commonly used micro-controllers in sensor networks. Their relevant features are mentioned in the following sub-sections.

2.5.1 PIC Series

The PIC series of micro-controllers [4] is manufactured by Microchip. They are widely used in embedded systems today. The features of the micro-controller are as follows.

Context Switching Features

- The PIC has no special features that help context switching but instead uses the common approach of saving the context i.e. registers, flags etc. each time the context is switched.

Interrupt Structure

- Interrupt structure is simple but powerful.
- Priority levels can be set for the interrupts (HIGH or LOW).
- The robust interrupt structure is suited to real-time interrupt driven applications.

ISA

- PIC instructions vary in number from about 35 instructions for the low-end PICs to about 70 instructions for the high-end PICs.
- PIC instruction size varies from 12 bits in the PIC12 series to 30 bits in the PIC30 series. All operands are 8 bit and so the PIC is called an 8-bit micro-controller.

- Most instructions are single cycle execution (4 clock cycles), with single delay cycles upon branches and skips.
- ISA is RISC-like but not exactly RISC as not all instructions are of fixed length and load-store architecture isnt followed.

Memory Architecture

- The PIC micro-controllers follow the Harvard architecture i.e. separate code and data space.
- PICs have a set of register files that function as general purpose RAM, special purpose control registers for on-chip hardware resources are also mapped into the data space.
- The data memory is divided into banks. The current bank is accessed in one instruction; otherwise the bank must be switched.
- The addressability of memory varies depending on device series, and all PIC devices have some banking mechanism to extend the addressing to additional memory. Later series of devices feature move instructions which can cover the whole addressable space, independent of the selected bank.

I/0 Features

- The PIC series of devices have I/O ports which can be used for transceiver and transducer/ADC interfacing.
- Some of the PIC devices even have on-chip ADCs that can be used to directly sample analog data from the transducers.
- The PIC series also have a UART though these arent required in the basic sensor node architecture, they can be used to interface to a PCs serial port on the nodes which behave as wired base stations.

- The PIC18 also has hardware which can carry out I2C or SPI.

Applicability of Low Power Techniques

- The PIC micro-controllers can operate over a wide frequency and voltage range which makes frequency scaling possible.
- The PIC12 and PIC16 series dont explicitly support frequency scaling but external hardware can be used for this purpose.
- The PIC18 series and above explicitly allows for frequency scaling by providing an internal RC oscillator that can be used as the clock. The RC oscillators frequency can be scaled.
- The PIC18 series also allows switching between external clock and the internal oscillator. Clock switching takes time and results in a delay of two old clock cycles and three new clock cycles.

Low Power Sleep Modes

- The sleep modes in the PIC series are extremely simplistic compared to some of the other processors (like TIs MSP430). As a matter of fact, the lower end PICs dont even provide sleep modes.
- The PIC16 series has only one sleep mode wherein the processor core and all other peripherals except the asynchronous timer is shut-off.
- The PIC18 series has two sleep modes, one in which only the core is shut-off and all clocks to peripherals keep running and the other in which all peripherals and the core is shut-off.

Pipelining

- PIC instructions generally take one machine/instruction cycle to execute i.e. 4 clock cycles.

- The execution of an instruction takes place in two phases: 1) Fetch and 2) Execute. This allows the PIC architecture to be pipelined.

- The pipeline is a two stage pipeline.

Shortcomings

- PIC microcontrollers have a very small set of instructions, leading some to consider them RISC devices. However, the PIC architecture does not reflect many of the advantages of RISC design. For example:

  a. PIC does not have a load-store architecture, as memory is directly referenced in arithmetic and logic operations

  b. It has a single working register, while RISC designs typically include 16 or more general purpose registers

  c. Its addressing modes are not orthogonal, since some instructions can address RAM or immediate data, while others can only use the working register

  d. Bank switching is required to access the entire RAM of many PIC devices, making the development of libraries of position-independent code complex and inefficient

  e. A stack cannot be implemented efficiently, so it is difficult to generate reentrant code

2.5.2 AVR Series

The AVR series of micro-controllers [4] follows a Harvard architecture single with an 8-bit RISC core running single cycle instructions. Particularly, the ATMEGA128L is a widely used micro controller in sensor nodes and is featured by several motes, including Berkeley Motes and Mica2/MicaZ [5].

Broad Device Classification

- tinyAVRs (e.g. all the ATTiny series)
    a. 1-8 KB program memory
    b. 8-20 pin package
    c. Limited peripheral set
- megaAVRs (e.g. all the ATmega series, including ATmega128L)
    a. 4-256 KB program memory
    b. 28-100 pin package
    c. Extended instruction set (instructions for multiply and handling larger program memory)
    d. Extensive peripheral set
- Application Specific AVRs
    a. AVRs with special features like LCD/USB controller, etc.

Interrupt Structure

- Powerful interrupt structure.
- The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles, the program vector address for the actual interrupt handling routine -is executed.

ISA

- The AVR ISA is more compact than most other 8-bit microcontrollers. The ATmega128 offers 133 powerful instructions
- Each instruction takes one or two 16-bit words

- Arithmetic operations work on registers R0-R31, but not directly on the RAM and take one clock cycle, excepting multiplication and word-wide addition which take two cycles

- RAM and I/O space can be accessed only by copying to or from registers. Indirect access (including optional post-increment, pre-decrement or constant displacement) is possible through registers X, Y, and Z (pointer registers)

- All accesses to RAM takes two clock cycles. Moving between registers and I/O is one cycle. Moving eight-bit or sixteen-between registers or constant to register is also one cycle. Reading program memory (LPM) takes three cycles

- A few AVR microcontrollers lack certain non-basic instructions like multiplication, extended loads/jumps/calls, long jumps and power control

Memory Architecture

- The AVR is Harvard architecture based with programs and data stored separately for performance and parallelism

- Flash, EEPROM and SRAM are all integrated on single chip, removing need for external memory

- The non-volatile Self-Programmable instruction flash memory (up to 256K) is used predominantly for storing program

- The data address space consists of the register file, I/O registers and the SRAM (up to 8K). The AVRs have 32 single-byte registers

- The AVR has memory-mapped I/O registers. The working registers occupy the first 32 memory addresses (000016 001F16) followed by 64 I/O registers (002016 005F16). Actual SRAM for data storage starts after the register section

I/O Features

- Bi-directional General Purpose I/O ports
- The AVRs have a built-in ADC and Analog Comparators
- On chip debugging (OCD) support through JTAG or debugWIRE on most devices. JTAG signals are multiplexed on GPIOs
- Serial Peripheral Interface and a Two-Wire Serial Interface for flexible communication
- Analog Comparators
- 10-Bit A/D Converters, with multiplex of up to 16 channels

Applicability of Low Power Techniques

- Low-voltage Devices Operating Down to 1.8V i.e. variable operating voltage
- Software Selectable Clock Frequency i.e. frequency scaling possible

Low Power Sleep Modes

- Six Power-Saving Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power down, Standby, and Extended Standby. User can tailor power consumption to applications requirement

Pipelining

- Each instruction takes one or two cycles and consists of the Fetch and the Execute cycles.
- Amtels AVRs have a single level pipeline design, i.e. next instruction is fetched as current one is executing

2.5.3 Comparison between AVR and PIC

- AVRs have non-banked access to data memory, whereas PICs require setting bank registers to access beyond 256 bytes of memory. Also, some AVRs support hooking up external SRAM in a way that allows the MCU to use it natively (rather than going through a series of port accesses)
- AVRs have 32 general purpose registers, the PIC only has one
- In AVRs that have SRAM (most of them), the stack is contained within SRAM instead of being limited to a built-in hardware stack. Conversely, with PIC, this is one less thing to worry about
- The ATmega and PIC18F have hardware multipliers, ATTiny and PIC16F do not
- The AVRs support a more generalized interrupt system, as opposed to the PIC high/low priority interrupt vectors
- Although PIC's clock speeds appear higher, the clock speed is divided by four to give the actual instruction rate

Chapter 3

MSP430 micro controller & development tools

3.1 MSP430 Architecture

The MSP430 incorporates a 16-bit RISC CPU, peripherals, and a flexible clock system that interconnect using a Von-Neumann common memory address bus (MAB) and memory data bus (MDB). Partnering a modern CPU with modular memory-mapped analog and digital peripherals, the MSP430 offers solutions for demanding mixed-signal applications.Key features of the MSP430x2xx family include[6]

- Ultra low-power architecture extends battery life
- 0.1-μA RAM retention , 0.8-μA real-time clock mode, 250-μA / MIPS active
- High-performance analog ideal for precision measurement
- 12-bit or 10-bit ADC 200 ksps, temperature sensor, Vref
- 12-bit dual-DAC
- Comparator-gated timers for measuring resistive elements
- Supply voltage supervisor

- 16-bit RISC CPU enables new applications at a fraction of the code size
- Large register file eliminates working file bottleneck
- Compact core design reduces power consumption and cost
- Optimized for modern high-level programming
- Only 27 core instructions and seven addressing modes
- Extensive vectored-interrupt capability
- In-system programmable Flash permits flexible code changes, field upgrades and data logging

Detail description of MSP430 architecture is given in Appendix A.

3.1.1 Low power operating modes

The MSP430 family was developed for ultra low-power applications and uses different levels of operating modes. The MSP430 operating modes, give advanced support to various requirements for ultra low power and ultra low energy consumption. This support is combined with an intelligent management of operations during the different module and CPU states. An interrupt event wakes the system from each of the various operating modes and the RETI instruction returns operation to the mode that was selected before the interrupt event.
The ultra-low power system design which uses complementary metal-oxide semiconductor (CMOS) technology, takes into account three different needs:

- The desire for speed and data throughput despite conflicting needs for ultra-low power
- Minimization of individual current consumption
- Limitation of the activity state to the minimum required by the use of low power modes

Low power mode control

Summary of various low power modes is given in I.

Table I: Various low power operating modes

SCG1	SCG0	OSCOFF	CPUOFF	Mode	CPU and Clock Status
0	0	0	0	active	CPU is active and all enabled clocks are active
0	0	0	1	LPM0	CPU and MCLK are disabled. SMCLK and ACLK are active
0	1	0	1	LPM1	CPU, MCLK and DCO osc. are disabled. DC Generator is disabled if DCO is not used for MCLK or SMCLK in active mode. SMCLK, ACLK are active
1	0	0	1	LPM2	CPU, MCLK, SMCLK and DCO osc. are disabled. DC generator remains enabled.ACLK is active
1	1	0	1	LPM3	CPU, MCLK, SMCLK and DCO osc. are disabled. DC generator disabled.ACLK is active
1	1	1	1	LPM4	CPU and all clocks disabled.

There are four bits that control the CPU and the main parts of the operation of the system clock generator:

- CPUOff
- OscOff
- SCG0
- SCG1

These four bits support discontinuous active mode (AM) requests, to limit the time period of the full operating mode, and are located in the status register. The major advantage of including the operating mode bits in the status register is that the present state of the operating condition is saved onto the stack during an interrupt service request. As long as the stored status register information is not altered, the processor continues (after RETI) with the same operating mode as before the interrupt event.

3.1.2 Watchdog Timer

The primary function of the watchdog timer (WDT) module is to perform a controlled system restart after a software problem occurs. If the selected time interval expires, a system reset is generated. If the watchdog function is not needed in an application, the module can be configured as an interval timer and can generate interrupts at selected time intervals.
Features of the watchdog timer module include:

- Four software-selectable time intervals
- Watchdog mode
- Interval mode
- Access to WDT control register is password protected
- Control of RST/NMI pin function
- Selectable clock source
- Can be stopped to conserve power

Detail description of Watchdog Timer is given in Appendix A.

3.2 MSP430 Development Tools

In this section, brief description of two MSP430 development toolkit is discussed.

3.2.1 eZ430-F2013 Development Toolkit

Description

Texas Instrument's new eZ430-F2013 is a comprehensive MSP430 development tool in a convenient USB stick form factor. It includes all the hardware and software necessary to evaluate the MSP430F2013 and develop a complete project. The eZ430-F2013 incorporates the IAR Embedded Workbench Integrated Development Environment (IDE), providing full emulation with the option of designing a stand-alone system or detaching the removable MSP430F2013 target board to integrate into an existing design.

Figure 3.1: MSP430F2013 Development Toolkit

The USB port provides enough power to operate the ultra-low power MSP430, so no external power supply is required. The MSP430F2013 includes 16-MIPS performance, a 16 bit Sigma Delta A/D converter, a 16-bit timer, Watchdog timer, brownout detector, a USI module supporting SPI and I2C, and five low power modes drawing as little as 0.1μA standby[7].

eZ430-F2013 Key Features

- eZ430-F2013 development tool including a USB debugging interface and detachable MSP430F2013 target board
- LED indicator
- Removable USB stick enclosure
- Debugging interface supports development with all MSP430F20xx devices
- Integrated IAR Kickstart user interface which includes an assembler, linker, simulator, source-level debugger and limited C-compiler
- Full documentation on CD-ROM

3.2.2 eZ430-RF2500 Development Toolkit

Description

The eZ430-RF2500 is a complete wireless development tool for the MSP430 and CC2500 that includes all the hardware and software required developing an entire wireless project with the MSP430 in a convenient USB stick. The tool includes a USB-powered emulator to program and debug your application in-system and two 2.4-GHz wireless target boards featuring the highly integrated MSP430F2274 ultra-low-power MCU. Projects may be developed and instantly deployed using the included battery expansion board and AAA batteries. All the required software is included such as a complete Integrated Development Environment and SimpliciTI, a propriety low-power star network stack, enabling robust wireless networks out of the box. The eZ430-RF2500 uses the MSP430F22x4 which combines 16-MIPS performance with a 200-ksps 10-bit ADC and 2 op-amps and is paired with the CC2500 multi-channel RF transceiver designed for low-power wireless applications.

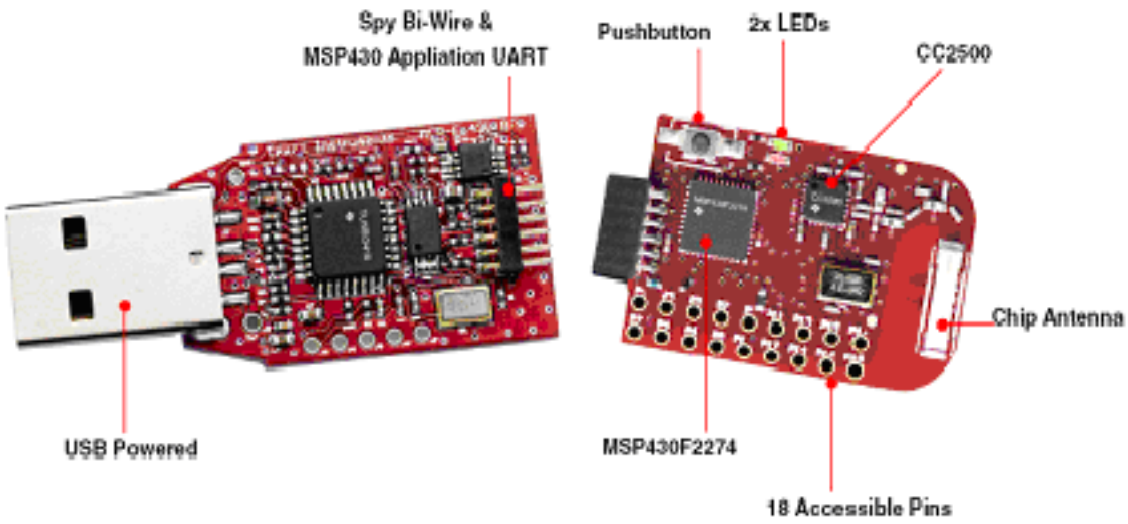


Figure 3.2: MSP430RF2500 Development Toolkit

The eZ430-RF emulator interface may be used with any Spy Bi-Wire enabled MSP430, such as the F22xx and F20xx series, and is fully compatible with the eZ430-F2013 and eZ430-T2012 target boards. The emulator interface can be used to download and debug your target applications, and can transmit serial data to your PC while in or out of a debug session[7].

eZ430-RF2500 Key Features

- eZ430-RF2500T, wireless target board
- MSP430F2274 - 32KB Flash, 1KB RAM, 1 USCI (UART, (2) SPI, I2C, IrDA), 10-bit 200 ksps, 2 Op Amps
- CC2500 - 2.4 GHz, ISM band multi-channel low power transceiver
- 2 LEDs and 1 push button
- eZ430-RF, debugging interface
- Supports MSP430 Application UART allowing serial communication to PC
- Supports eZ430-T2012 and eZ430-RF2500T target boards

- Removable USB stick enclosure
- Battery expansion board with 2 AAA batteries
- SimpliciTI, low power network stack
- An assembler, linker, source-level debugger and limited C-compiler

Chapter 4

Position Based Routing

4.1 Unit Graph Representation of Multi-hop Wireless Network

The Unit Graph Model is the basic widely used graph-theoretical model for all networks (Figure 4.1), where R is the transmission radius and is equal for all the nodes inside the network. This model varies depending on the obstacles, minimum power graphs of the unit graph where each node has its own transmission radius and unidirectional links or not.

4.2 Position Based Routing Taxonomy

The desirable qualitative properties include:

Loop Freedom classified as having loop free property or not.

Distributed Operation resemble greedy algorithms where simple local behavior may achieve a desired global objective.

Path Strategy single path/multi-path strategy.

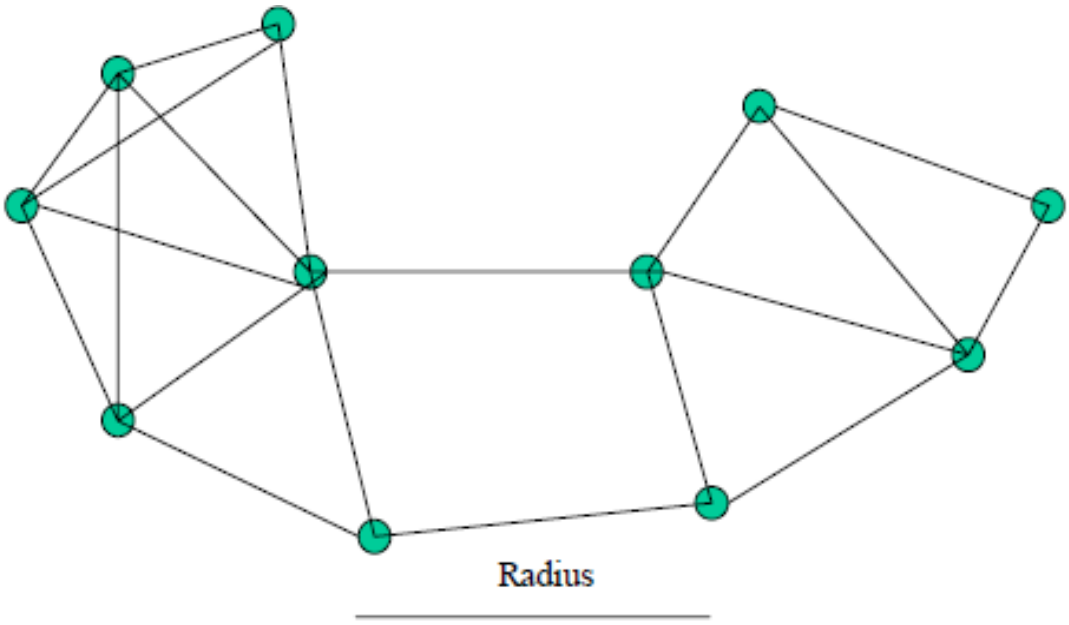


Figure 4.1: Unit Graph Representation of Multi-hop Wireless Network[8]

Metrics hop count (the amount of nodes of the transmission on the route from source to destination) used most widely; power metric that depends on the distance between nodes may substitute the constant metric if the nodes can adjust their transmission power; the cost metric aims to maximize the number of routing tasks that network can perform.

Memorization memorizing the route or past traffic is sensitive, its better to avoid memorizing past traffic at any node as possible, and the memorization is needed for short period of time while ongoing task is in progress.

Guaranteed Message Delivery The goal is to delivery the message and gains the best assurance. Wireless networks often use the single frequency communication model, the broadcasting is applied so that the message sent is heard by all other neighbors within the transmission radius of the sender.

Scalability simplified criterion is applied that a routing scheme is scalable if it is loop-free, localized and single-path. The message delivery can be guaranteed in

the static case, but hard to handle the loops in the case of node mobility. The loops are named by their positions.

Robustness the accuracy of destination position is important. This talk only concentrates on the fixed node destination. [8]

4.3 Geographic Routing

Geographic routing is based on two principal assumptions:

- It is assumed that every node knows it own and its network neighbors positions.
- The source of a message is assumed to be informed about the position of the destination.

This is a new technique for localized broadcasting of queries in geo-aware sensor networks; it makes use of the existing query routing tree and does not involve the creation of any additional communication channels. Geographic routing is very interesting because it operates without any routing tables. Furthermore, once the position of the destination is know, all operations are strictly local, that is, every node is required to keep track only of its direct neighbor.

4.3.1 Definition

Let G = (V,E) be a Euclidean graph. The task of a geographic ad hoc routing algorithm A is to transmit a message from a source s $\in$ V to a destination t $\in$ V by sending packets over the edges of G while complying with the following conditions:

- All nodes v $\in$ V know their geographic positions as well as the geographic positions of all their neighbors in G.
- The source s is informed about the position of the destination t.

- The control information which can be stored in a packet is limited by O(log n) bits, that is, only information about a constant number of nodes is allowed.
- Except for the temporary storage of packets before forwarding, a node is not allowed to maintain any information.

Description of different types of Geographic routing algorithm is given in following subsections.

4.3.2 Greedy Routing(GR)

The Greedy routing route the message to another node, that near to the destination node is. It can be formulated as follow:

a. Packet at s

b. Give the packet to the node nearest to the destination t

c. Repeat step 2 until the packet reach t, other until the packet reach a node v without any neighbor closer to t as itself.

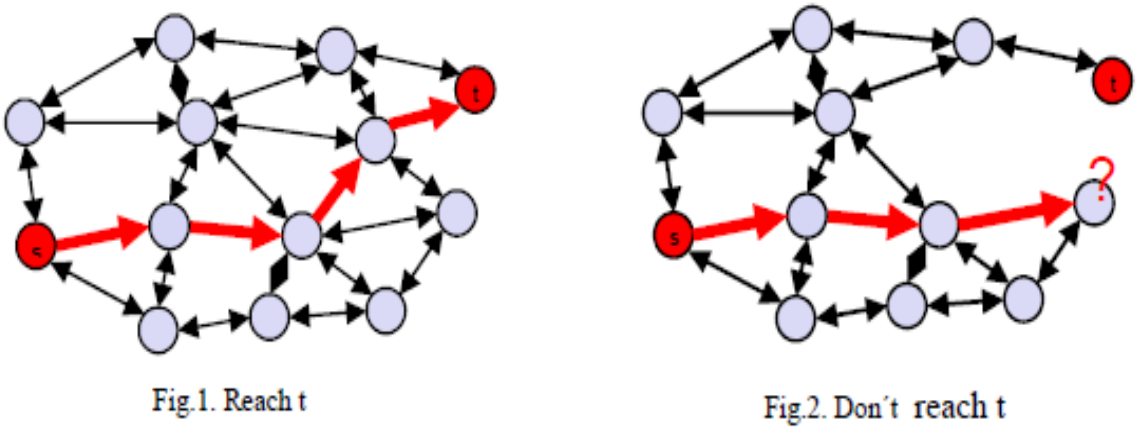


Fig.1. Reach t

Fig.2. Don't reach t

Figure 4.2: Two different scenario in geographic routing[9]

As shown in figure 4.2, we can find two possibilities. The first one is that the packet reaches t, and the second one is that the packet never reaches t. If the packet reaches t, than with cost O(d^2), where d := $|\bar{st}|$ st denote the Euclidean distance between s and t.

4.3.3 Face Routing(FR)

Face Routing use the concept of faces, contiguous regions separated by the edges of a planar graph that is a graph containing no two intersecting edges. The algorithm explores face boundaries employing the right hand rule.

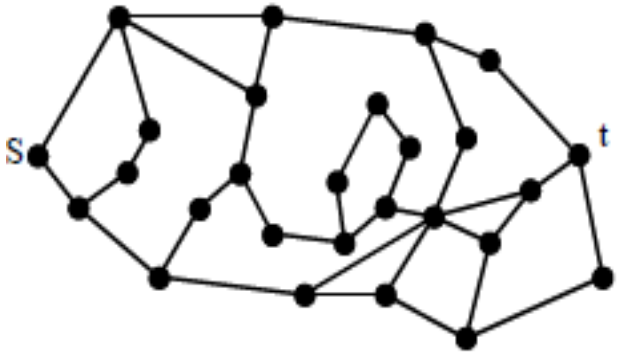


Figure 4.3: Face Routing

The algorithm takes the way around a face, and then it keeps track of the points where it crosses the line st connecting s and t. Having completely surrounded a face, it returns to the point closest to t. And now it explores the next face closer to t. If s and t are connected, FR finds always a path to t.
The total cost of Face routing us O (n), because:

- Each face is explore at most one

    a. Each edge is traversed at most four times

- There are at most 3n-6 edges (Euler)

There are too some disadvantage with FR; if source and destination are closed to each other, FR can take O (n) steps. Then we need some optimal algorithm.

4.3.4 Adaptive Face Routing

The main problem of face routing is that it must explore all the faces. It is thus impossible to bound the cost of this algorithm by the cost of an optimal path between source and destination. The Adaptive Face Routing tries to minimize the step between the source and the destination. This advantage consists in limited the cost with respect to the length of the shortest path between the source and the destination.

Description

Start with a small searchable area, explore the face (employing the right hand rule), when the path does not exist then change the direction. Find the node closer to the destination (t), and repeat the process. When there is not path to another node closer to t than come back and make the area bigger, and repeat all the process until reach t. AFR algorithm finds t after $O(c^2)$ steps, c is the cost of an optimal path from source to destination. Figure 4.4 explores steps briefly.

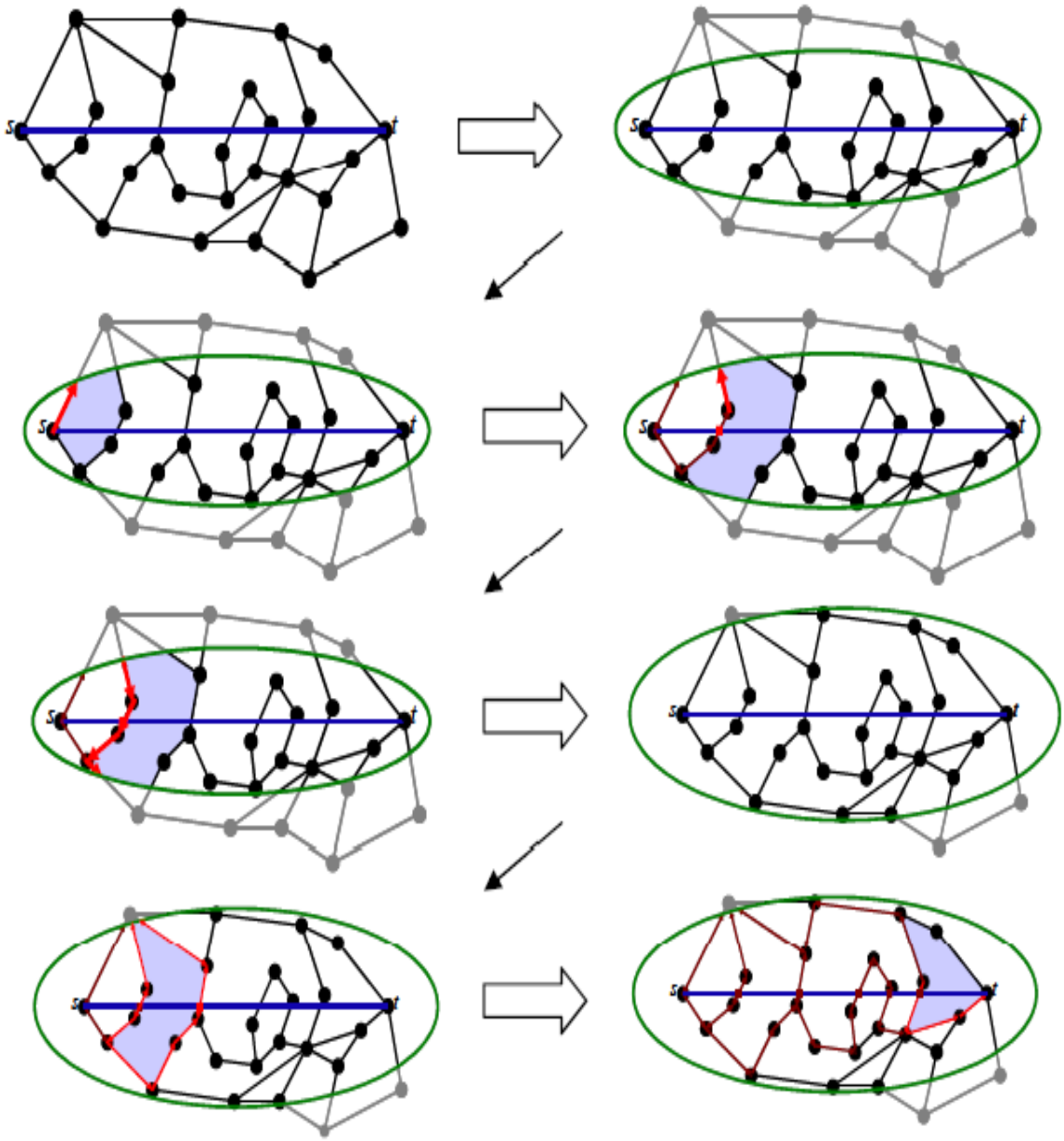


Figure 4.4: Adaptive Face Routing

Chapter 5

Implementation, Results & Analysis

5.1 Texas Instrument Virtual Lab

Virtual Lab Technology allows instant, easy access to Hardware and Software using a standard web browser for evaluation, demonstration and training. Virtual Labs contain real hardware and fully functional pre-configured software. The snapshot of Texas Instrument Online Virtual Lab is given in figure 5.1.

5.1.1 IAR Embedded Workbench for MSP430

IAR Embedded Workbench is a set of highly sophisticated and easy-to-use development tools for embedded applications. It integrates the IAR C/C++ Compiler, assembler, linker, librarian, text editor, project manager, and C-SPY Debugger in an integrated development environment (IDE). With its built-in chip-specific code optimizer, IAR Embedded Workbench generates very efficient and reliable FLASH/ PROMable code for the MSP430 micro controller. In addition to this solid technology, IAR Systems also provides professional worldwide technical support.

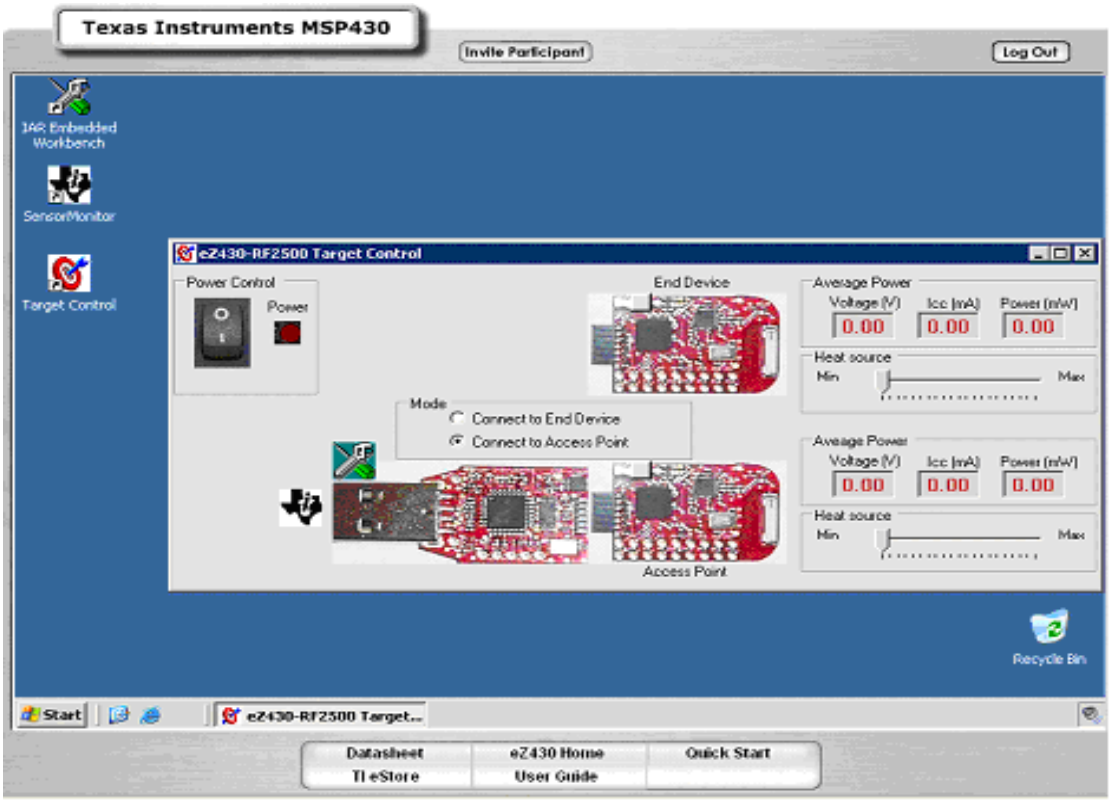


Figure 5.1: Texas Instrument Online Virtual Lab[10]

Benefits

IAR Embedded Workbench offers the same intuitive user interface regardless of which micro controller you have chosen to work with general and target-specific support for each device.

Every IAR C/C++ Compiler contains both generic global optimizations as well as low-level chip-specific optimizations that ensure a small code size while taking advantage of all the specific features of your selected device.

Reuse of code and migration to new micro controller architectures is made easy as each IAR C/C++ Compiler uses the same naming convention.

Whether you have a tight project schedule or are just eager to get started, IAR Embedded Workbench contains configuration files, code examples and template projects to get you going fast.

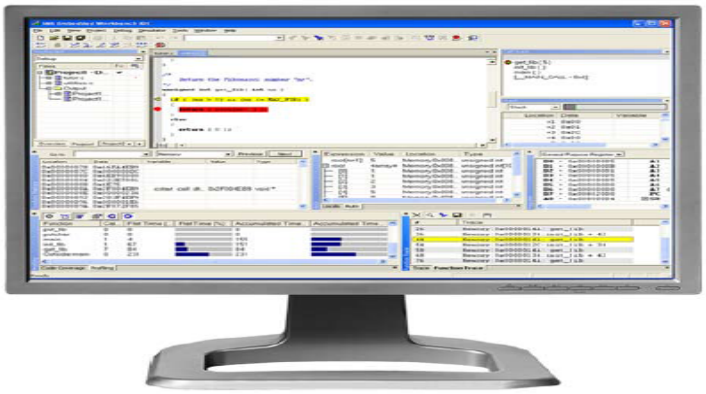


Figure 5.2: IAR Embedded Workbench[11]

5.1.2 Intrinsic Functions used for MSP430

The header file <inmsp430.h> defines a number of intrinsic functions of general use for the MSP430 processor.

Table I: Various Intrinsic Functions

Intrinsic Functions	**Description**
__bic_SR_register	Clear bits in status register
__bic_SR_register_on_exit	Clear bits in stacked status register
__low_power_mode_off_on_exit	Enter active mode when interrupt returns
__bis_SR_register	Set bits in status register
__bis_SR_register_on_exit	Set bits in stacked status register
__disable_interrupt	Disable global interrupts
__enable_interrupt	Enable global interrupts
__low_power_mode_0	Enter low power mode 0
__low_power_mode_1	Enter low power mode 1
__low_power_mode_2	Enter low power mode 2
__low_power_mode_3	Enter low power mode 3
__low_power_mode_4	Enter low power mode 4

5.2 Proposed Design Methodology

In typical embedded systems, the main processor derives power from either AC/DC wall adapter or from a battery, whereas the micro controller is powered from either of the above sources and - in their absence - from a coin cell to keep it powered on at all times. Coin cells used in embedded systems are of low capacity and usually not rechargeable, but they must be capable of powering up the micro controller for the expected portable product life cycle.

When the rest of the system is switched off, only the watchdog timer is expected to be running. Since the micro controller handles the watchdog timer and other user interfaces, it makes sense to switch off all its peripherals except the watchdog timer and to configure it to a lower power consuming mode.

When the system is powered up or when system RESET is de-asserted, the micro controller has to change back to its normal state where all its peripherals are active and the CPU can work at maximum computational power. Thus, the micro controller has to be put in normal state. Consider that the micro controller is running from a coin cell (non rechargeable). Then the lifetime of the coin cell is calculated using following equation.

Coin cell life time(in hr) = Battery capacity(mAh) / MCU current consumption(mA)

Consider the timer module of an micro controller used for Real Time Clock(RTC) in its normal state with millisecond resolution; i.e. the timer interrupt occurs every 1 ms to update the RTC value. When the micro controller switches to its low power state, all its peripherals are off except for the timer required to keep the RTC value updated.

T1 = Amount of time Micro controller is present in low power state
T2 = Time taken to update the RTC value

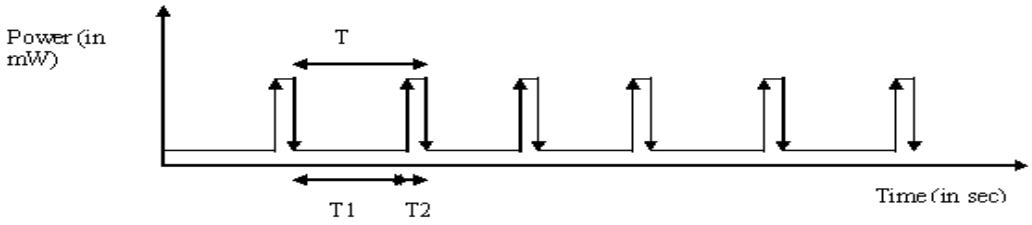


Figure 5.3: Micro controller Timer Interrupt

T = Timer resolution / Time interval between two successive timer interrupts.
I_{low} = Micro controller low power state current consumption
I_{high} = Current consumption by micro controller in active state

The average current consumption is given by the following equation:
Average current consumption by MCU
= ((MCU low power state current consumption * Amount of time the MCU is present in its low power state) + (Current consumption by the MCU to update the RTC value * Time taken to update RTC value)) / (Timer resolution)
= ((I_{low} * T1) + (I_{high} * T2)) / T

If the timer is allowed to run at a second resolution that is, a resolution of one second) when the MCU switches to its low power state, then the average current consumption by MCU is as follows:
Average current consumption by MCU
= ((I_{low} * (T-T2)) + (I_{high} * T2)) / T

5.2.1 Proposed Geographic Distance Routing Algorithm

The Geographic distance routing algorithm have five major functions to be performed. According to these functions, most suitable power mode has to be chosen in order to reduce the overall power consumption of processor. These functions are described below.

- Neighbor Discovery
- Computation of Euclidian Distance & Selection of node nearest to source node based on shortest Euclidian Distance
- Control packet processing and transmission of control as well as data packet
- Receiving Reply packet from Destination node
- Processing of Received packet

Neighbor Discovery requires only transceiver to be on so the processor has to be in the low power mode 3. Computation of Euclidian Distance & Selection of node nearest to source node based on shortest Euclidian Distance requires some computation so there is a need for processor to be in active state. Therefore the processor has to be in the active mode. Control packet processing and transmission of control as well as data packet requires both processor and transceiver to be in active state so the processor has to be in the active mode. Receiving Reply packet from Destination node requires only transceiver to be active so the processor has to be in the low power mode 3. Processing of Received packet requires processor to be active so the processor has to be in the active mode.

The geographic distance routing algorithm has following functions:

Set Degree This function gives the number of nodes connected to the current node.

Calculate Radius This function finds neighbor node for the current node.

Set Edge This function establish wireless link between current node and neighbor node.

GEDIR_Path This function determines path between source node and destination node.

Send_packet This function sends packet from one node to another.

Receive_packet This function receive packet from one node to another.

5.3 Results

Watchdog timer is used as periodic timer that will generate periodic interrupt at an interval of 1 sec. Toggle rate is exactly 1sec based on 32kHz ACLK WDT clock source. In this experiment, the WDT is configured to divide 32768 Hz watch-crystal 2^{15} by 2^{13} with an Interrupt Service Routine triggered @ 4Hz.

The power consumption without using MSP430 micro controller is given in figure 5.4.

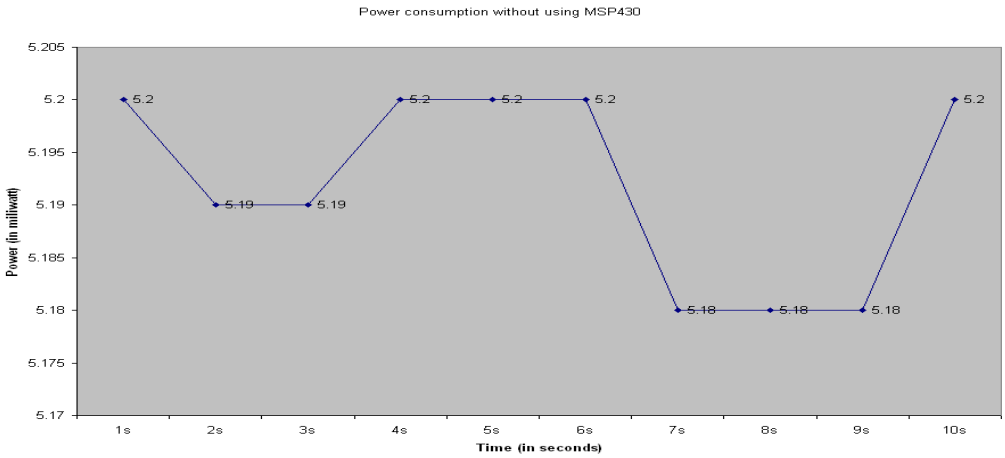


Figure 5.4: Power consumption without using MSP430 micro controller

As shown in figure 5.4, power consumption of the processor is steady. Although there is no need for all the components of micro controller, the processor will consume maximum power (here in the range of 5.18mW to 5.20mW).

The power consumption with using MSP430 micro controller is given in figure 5.5.

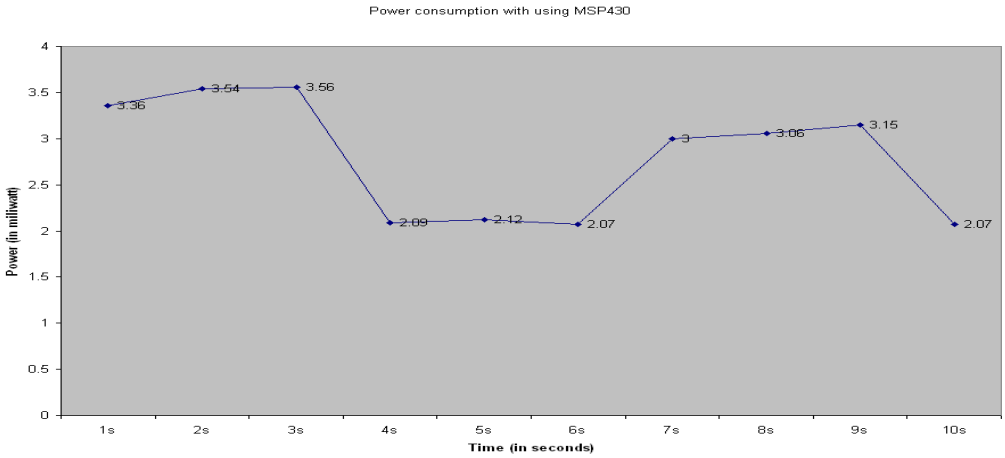


Figure 5.5: Power consumption with using MSP430 micro controller

As shown in figure 5.5, during first 3 seconds, processor is busy with control packet

processing as well as transmission of control packet, so it is consuming more power i.e. in the range of 3.36mW to 3.56mW. During 4 to 6 seconds, processor is sending and receiving packets from neighbor, so only transceiver needs to be on so it is consuming much less power in the range of 2.07mW to 2.12mW. During 7 to 9 seconds, processor is only doing control packet processing so it is consuming power in the range of 3mW to 3.15mW. During 10^{th} second, processor is either receiving or sending packet, so only transceiver needs to be on so it will consume much less power i.e 2.07mW.

5.3.1 Result Analysis

The power consumption without using MSP430 micro controller is **5.20mW** and the power consumption with using MSP430 micro controller is **2.07mW**. So by using ultra low power micro controller, power consumption of processor is reduced by 60%. But one drawback of this proposed method is that it will decrease the throughput when micro controller is used in low power mode.

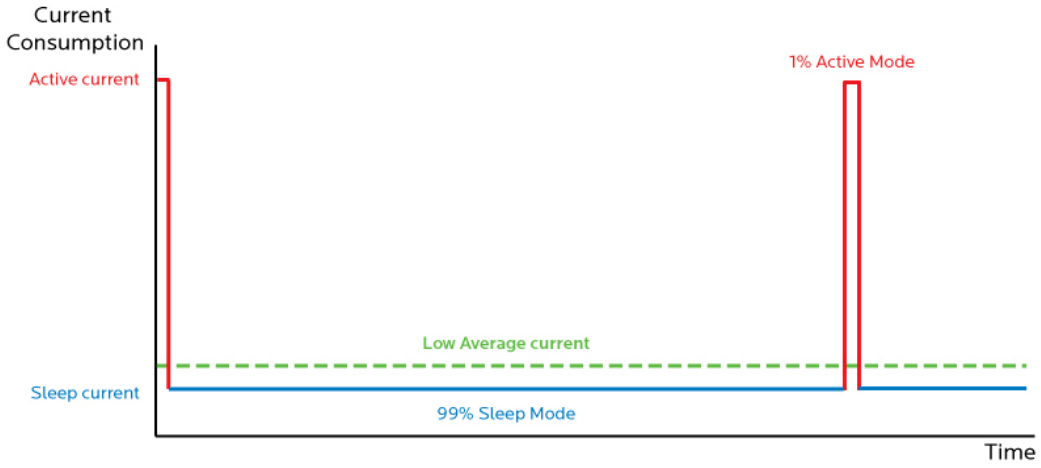


Figure 5.6: Current consumption of MSP430 using watchdog timer in timer mode

Lower the operating frequency is important to reduce dynamic power consumption. If the frequency is lowered, the application spends more time in active mode and less in the battery saving static mode. This increases average power consumption and decreases battery life. So if the processor is frequently switching from active mode to

low power mode, then throughput is decreased as compared to running processor in only active mode. This scenario is shown in figure 5.6.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Sensor networks have emerged as a revolutionary technology for querying the physical world and hold promise in a wide variety of applications. However, the extremely energy constrained nature of these networks necessitate that their design and operation be done in an energy-aware manner, enabling the system to dynamically make tradeoffs between performance, fidelity, and energy consumption.

The power optimization for a wireless sensor network can be achieved either by using efficient less power consuming hardware or by developing optimized software code. The MSP430 micro controller has large number of ultra low power modes which has very less power consumption in sleep mode. By using ultra low power micro controller, power consumption of processor is reduced by 60%.

6.2 Future Scope

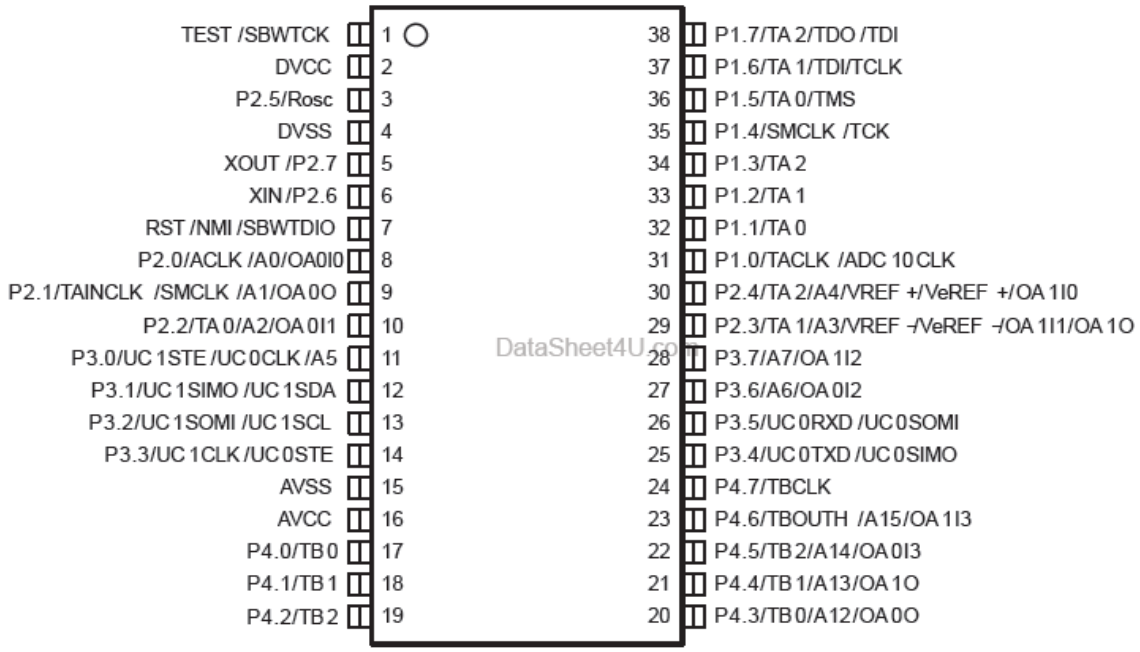
Currently, routing algorithm is implemented using watchdog timer in timer mode of MSP430 ultra low power micro controller but it drastically reduce the throughput. So there is a chance to improve the throughput and for that there is an alternative available that will use watchdog timer in watchdog mode so whenever there will be

a need for processor to be in active state then only processor will be in active state otherwise it will be in low power mode. So this method will improve the throughput.

Appendix A

MSP430 Micro controller

A.1 MSP430x22x4 device pinout, DA Packages



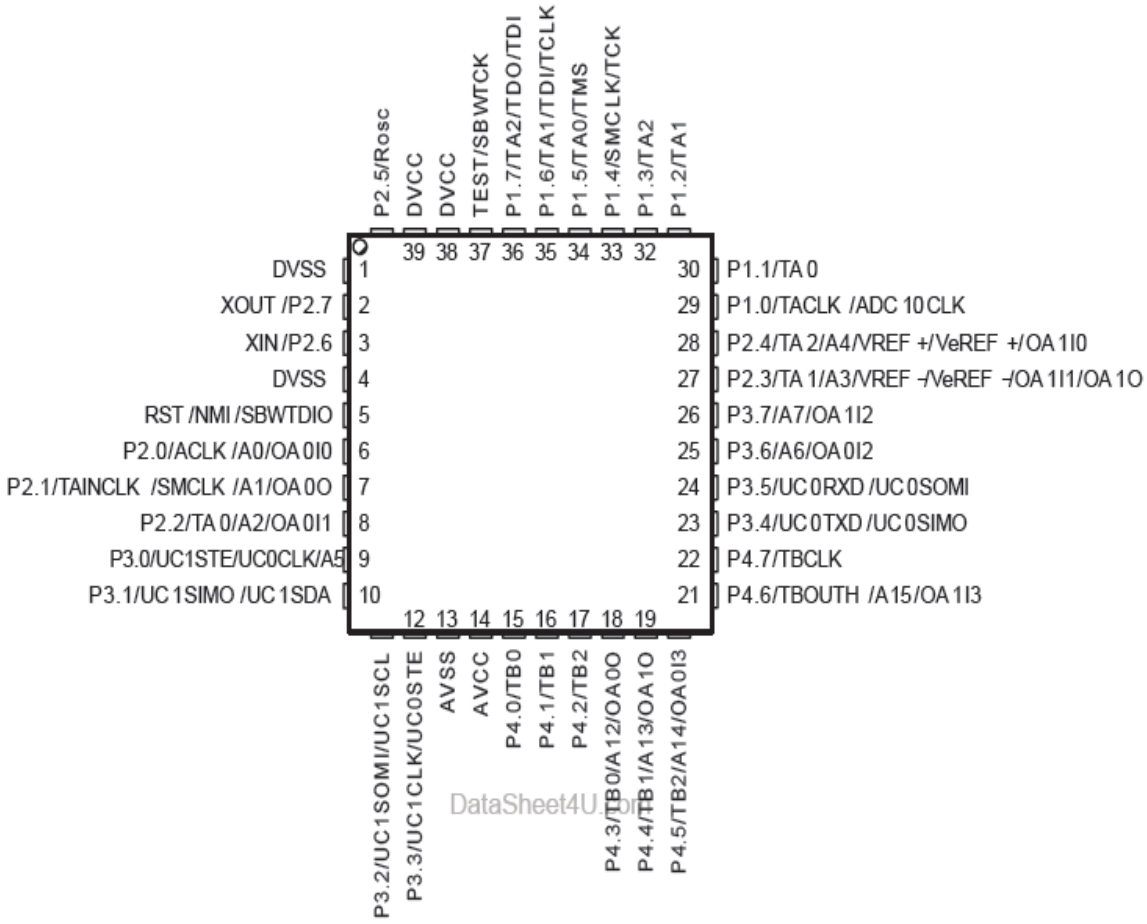
A.2 MSP430x22x4 Micro controller Features

MSP430x22x4
MIXED SIGNAL MICROCONTROLLER

SLASXXX – NOVEMBER 2005

- Low Supply Voltage Range 1.8 V to 3.6 V
- Ultralow-Power Consumption
  - Active Mode: 250 μA at 1 MHz, 2.2 V
  - Standby Mode: 0.7 μA
  - Off Mode (RAM Retention): 0.1 μA
- Ultrafast Wake-Up From Standby Mode in less than 1 μs
- 16-Bit RISC Architecture, 65 ns Instruction Cycle Time
- Basic Clock Module Configurations:
  - Internal Frequencies up to 16MHz with 4 Calibrated Frequencies to ±1%
  - Internal very low power LF oscillator
  - 32-kHz Crystal
  - High-Frequency Crystal up to 16MHz
  - Resonator
  - External Digital Clock Source
  - External resistor
- 16-Bit Timer_A With Three Capture/Compare Registers
- 16-Bit Timer_B With Three Capture/Compare Registers
- 10−Bit, 200ksps A/D Converter With Internal Reference, Sample−and−Hold, Autoscan, and Data Transfer Controller
- Two Configurable Operational Amplifiers
- Universal Serial Communication Interface (USCI) with
  - Enhanced UART supporting auto−baudrate detection (LIN)
  - IrDA Encoder and Decoder
  - Synchronous SPI
  - I2C™
- Brownout Detector
- Serial Onboard Programming, No External Programming Voltage Needed Programmable Code Protection by Security Fuse
- Bootstrap Loader
- On Chip Emulation Module
- MSP430F22x4 Family Members Include:
  MSP430F2234: 8KB + 256B Flash Memory 512B RAM
  MSP430F2254: 16KB + 256B Flash Memory 512B RAM
  MSP430F2274: 32KB + 256B Flash Memory 1KB RAM
  Available in a 38-Pin Plastic Small-Outline Thin (TSSOP) Package, and 40-Pin QFN Package
- For Complete Module Descriptions, Refer to the *MSP430x2xx Family User's Guide*

A.3 MSP430x22x4 device pinout, RHA Packages



A.4 MSP430 Architecture

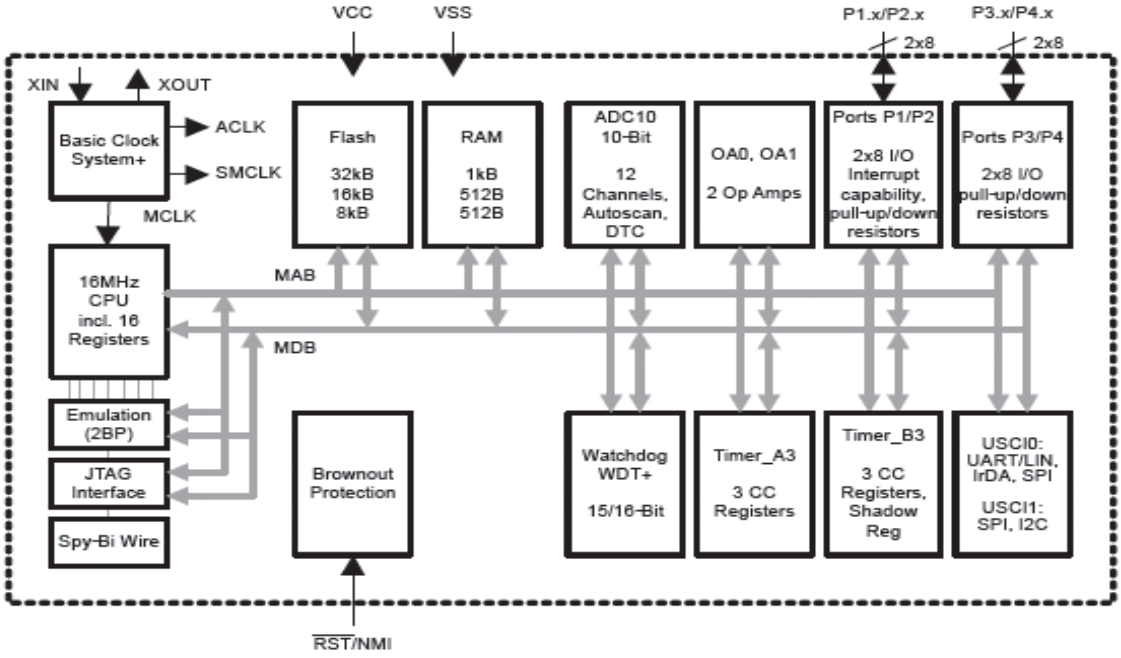


Figure A.1: MSP430x22x4 Architecture [12]

A.4.1 Address Space

The MSP430 von-Neumann architecture has one address space shared with special function registers (SFRs), peripherals, RAM, and Flash/ROM memory as shown in A.2. Code access are always performed on even addresses. Data can be accessed as bytes or words. The addressable memory space is 64 KB with future expansion planned.

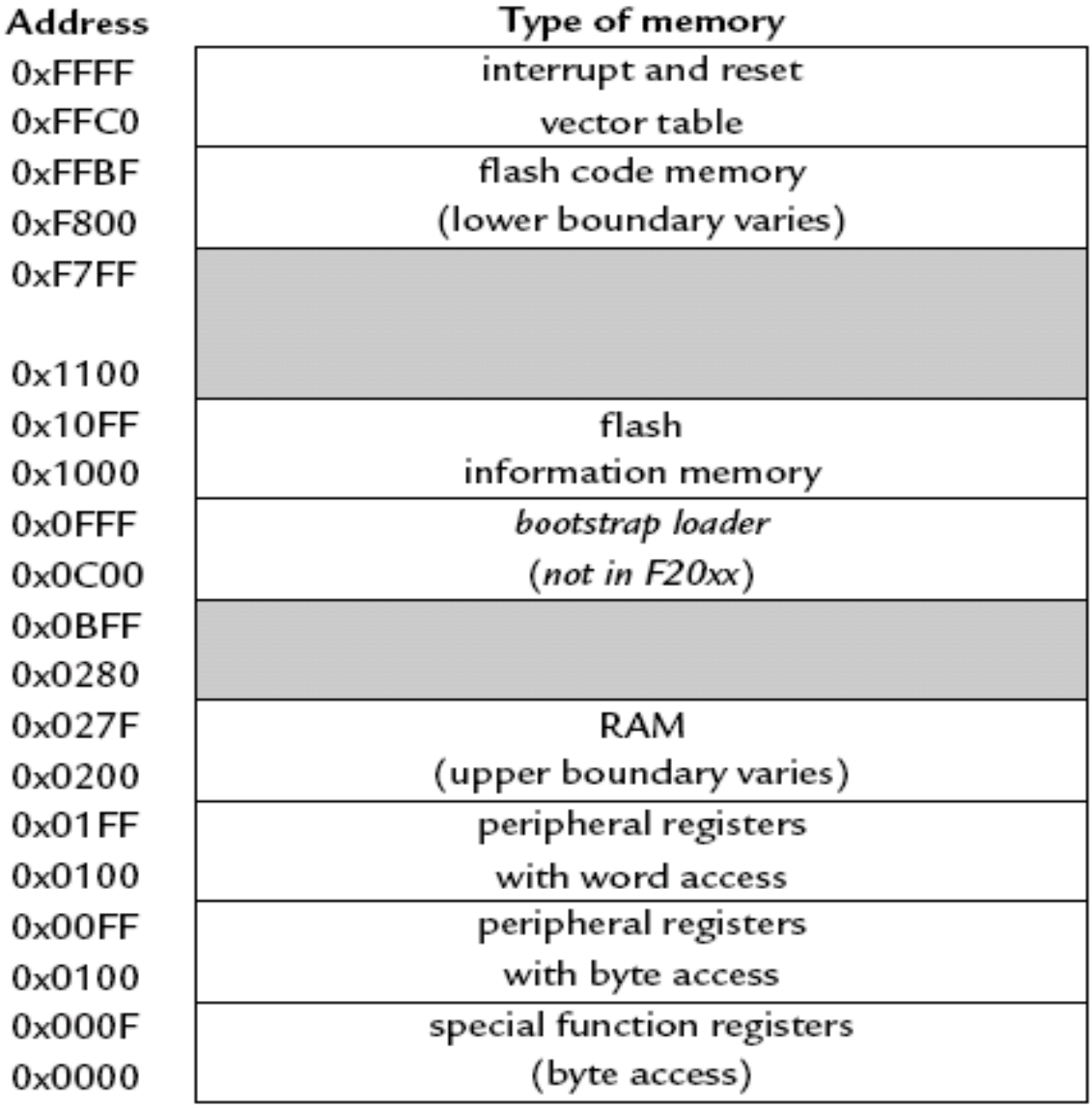


Figure A.2: Memory Map[13]

Flash/ROM

The start address of Flash/ROM depends on the amount of Flash/ROM present and varies by device. The end address for Flash/ROM is 0FFFFh. Flash can be used for both code and data. Word or byte tables can be stored and used in Flash/ROM

without the need to copy the tables to RAM before using them. The interrupt vector table is mapped into the upper 16 words of Flash/ROM address space, with the highest priority interrupt vector at the highest Flash/ROM word address (0FFFEh).

RAM

RAM starts at 0200h. The end address of RAM depends on the amount of RAM present and varies by device. RAM can be used for both code and data.

Peripheral Modules

Peripheral modules are mapped into the address space. The address space from 0100h to 01FFh is reserved for 16-bit peripheral modules. These modules should be accessed with word instructions. If byte instructions are used, only even addresses are permissible, and the high byte of the result is always 0. The address space from 010h to 0FFh is reserved for 8-bit peripheral modules. These modules should be accessed with byte instructions. Read access of byte modules using word instructions results in unpredictable data in the high byte. If word data is written to a byte module only the low byte is written into the peripheral register, ignoring the high byte.

Special Function Registers(SFRs)

Some peripheral functions are configured in the SFRs. The SFRs are located in the lower 16 bytes of the address space, and are organized by byte. SFRs must be accessed using byte instructions only. See the device-specific data sheets for applicable SFR bits.

A.4.2 Interrupts

The interrupt priorities are fixed and defined by the arrangement of the modules in the connection chain. The nearer a module is to the CPU/NMIRS, the higher the priority. Interrupt priorities determine what interrupt is taken when more than one

interrupt is pending simultaneously.
There are three types of interrupts.

System Reset The system reset circuitry sources both a power-on reset (POR) and a power-up clear (PUC) signal. Different events trigger these reset signals and different initial conditions exist depending on which signal was generated.

(Non)-Maskable Interrupts (NMI) (Non)-maskable NMI interrupts are not masked by the general interrupt enable bit (GIE), but are enabled by individual interrupt enable bits (NMIIE, ACCVIE, OFIE). When a NMI interrupt is accepted, all NMI interrupt enable bits are automatically reset. Program execution begins at the address stored in the (non)-maskable interrupt vector, 0FFFCh.
A (non)-maskable NMI interrupt can be generated by three sources:

- An edge on the RST/NMI pin when configured in NMI mode
- An oscillator fault occurs
- An access violation to the flash memory

Maskable Interrupts Maskable interrupts are caused by peripherals with interrupt capability including the watchdog timer overflow in interval-timer mode. Each maskable interrupt source can be disabled individually by an interrupt enable bit, or all maskable interrupts can be disabled by the general interrupt enable (GIE) bit in the status register (SR).

A.4.3 CPU Introduction

The CPU incorporates features specifically designed for modern programming techniques such as calculated branching, table processing and the use of high-level languages such as C. The CPU can address the complete address range without paging. The CPU features include:

- RISC (Reduced Instruction Set for Computer) architecture with 27 instructions and 7 addressing modes.

- Orthogonal architecture with every instruction usable with every addressing mode.
- Full register access including program counter, status registers, and stack pointer.
- Single-cycle register operations.
- Large 16-bit register file reduces fetches to memory.
- 16-bit address bus allows direct access and branching throughout entire memory range.
- 16-bit data bus allows direct manipulation of word-wide arguments.
- Constant generator provides six most used immediate values and reduces code size.
- Direct memory-to-memory transfers without intermediate register holding.
- Word and byte addressing and instruction formats.

RISC

Reduced Instruction Set Computer is a type of microprocessor architecture that utilizes a small, highly-optimized set of instructions, rather than a more specialized set of instructions often found in other types of architectures.
The simplest way to examine the advantages and disadvantages of RISC architecture is by contrasting it with it's predecessor: CISC (Complex Instruction Set Computers) architecture.

A.5 Watchdog Timer Operation

The WDT module can be configured as either a watchdog or interval timer with the WDTCTL register. The WDTCTL register also contains control bits to configure the

Table I: Difference between RISC and CISC Architecture

CISC	**RISC**
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

RST/NMI pin. WDTCTL is a 16-bit, password-protected, read/write register. Any read or write access must use word instructions and write accesses must include the write password 05Ah in the upper byte. Any write to WDTCTL with any value other than 05Ah in the upper byte is a security key violation and triggers a PUC system reset regardless of timer mode. Any read of WDTCTL reads 069h in the upper byte.

A.5.1 Watchdog Timer Counter

The watchdog timer counter (WDTCNT) is a 16-bit up-counter that is not directly accessible by software.

Table II: Watchdog Timer Registers

Register	**Short Form**	**Register Type**	**Address**	**Initial State**
Watchdog Timer Control Register	WDTCTL	Read/Write	0120h	06900h with PUC
SFR Interrupt Enable Register 1	IE1	Read/Write	0000h	Reset with PUC
SFR Interrupt Flag Register 1	IFG1	Read/Write	0002h	Reset with PUC

The WDTCNT is controlled and time intervals selected through the watchdog

timer control register WDTCTL. The WDTCNT can be sourced from ACLK or SMCLK. The clock source is selected with the WDTSSEL bit.

References

[1] S. LINDSEY and K. M. SIVALINGAM, *Power Optimization in Routing Protocols for Wireless and Mobile Networks*, ch. 19, p. 17.

[2] S. J. Ali and P. Roy, "Energy saving methods in wireless sensor networks," Tech. Rep. IDE0814, School of Information Science, Computer and Electrical Engineering Halmstad University, Box 823, S-301 18 Halmstad, Sweden, May 2008.

[3] S. Ahmed, "Trlabs report on current researches on sensor networks," tech. rep., May 2004.

[4] D. Singh, "Micro-controller for sensor networks," Master's thesis, Indian Institue of Technology, Kharagpur, May 2008.

[5] Website - Crossbow Inc. `www.xbow.com`.

[6] A. Punit and K. Kashyap, "MSP430 mcu based generic sensor appplication," tech. rep., Goverment Engineering College, Gandhinagar, Near Visat Petrol Pump, Chandkheda, Gandhinagar, May 2006.

[7] Website - Texas Instruments Micro controller MSP430. `www.ti.com/msp430`.

[8] L. Zhang, "Algorithms for ad hoc networking," seminar talk report, September 2003.

[9] J. Ariza, "Geographic routing protocol vs. table driven protocols," seminar routing algorithm.

[10] Website - Texas Instrument Online Virtual Lab. `http://www.techonline.com/product/virtuallab`.

[11] Website - IAR Embedded Workbench for MSP430. `http://www.iar.com`.

[12] Texas Instruments, *MSP430x22x4 MIXED SIGNAL MICROCONTROLLER*, November 2005.

[13] J. Davis, *MSP430 Microcontroller Basics*. Newnes, 2008.

Index