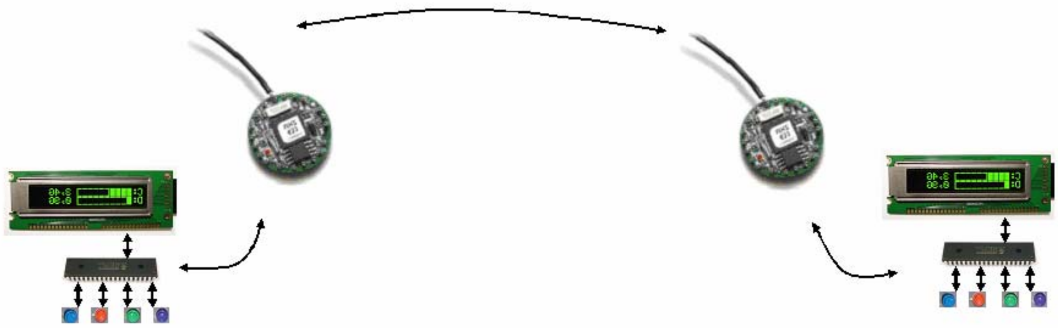
master's thesis, certec, lth number 1:2006

Vipul Surve

A Wireless Communication Device for Short Messages



Certec
Division of Rehabilitation Engineering Research
Department of Design Sciences
Faculty of Engineering
Lund University

Department of Electroscience
Department of Electroscience
Faculty of Engineering
Lund University

Sammanfattning

Det är viktigt att få veta en familjmedlems hälsotillstånd när han eller hon är sjuk. I sådana fall behöver standardmeddelanden utbytas som t ex “Hur mår du?”, “Jag mår bra”, “Jag är väldigt sjuk”, “Ring mig genast” eller “Jag mår bättre nu”. Det är också viktigt att ha en kommunikationslänk som är tyst, eftersom personen kanske sover. Meddelanden skulle också kunna utbytas mellan vänner eller släktingar, t ex ”Hur är det?”, ”Vill du ta en sväng?”, ”Jag har tråkigt”, ”Jag sticker nu” osv. I sådana fall finns det ett behov av ett snabbt utbyte av korta meddelanden. Vissa standardmeddelanden upprepas ofta i sådana konversationer.

Syftet med detta examensarbete är att utveckla mjukvaran för trådlösa sensorer, som kan användas för att skicka och ta emot korta meddelanden. För att göra enheterna portabla, användes trådlösa sensorer för att länka samman dem. Utöver detta har en studie gjorts för att visa vilken typ av användargränssnitt som skulle kunna användas tillsammans med dessa trådlösa sensorer.

Vid kommunikation mellan två sensorer krävs ofta en tredje sensor för att styra det trådlösa sensornätverket. Denna tredje sensor, som oftast kallas för “nod 0” eller “basstation”, måste vara kopplad till en dator. Huvuduppgiften är att hitta ett sätt att ta bort basstationen så att två trådlösa sensorer kan kommunicera med varandra utan hjälp av en basstation. På detta sätt minskar man den infrastruktur som krävs för nätverket.

En mjukvara har utvecklats som gör det möjligt för två eller fler sensorer att kommunicera med varandra utan basstation. Flera tester har gjorts med mjukvaran för att testa dess funktionalitet, det trådlösa nätverkets räckvidd och möjligheten att använda fler än två sensorer i ett nätverk.

En modell av en handburen enhet som kan användas i ett sådant nätverk togs också fram. Intervjuer gjordes med experter för att studera användbarheten av en sådan enhet och möjliga användargränssnitt.

Slutsatsen är att basstationen kan elimineras i ett trådlöst sensornätverk och att det ändå är möjligt att kommunicera mellan två eller flera enheter. Det är också möjligt att bygga en handburen enhet baserad på modellen så att två användare kan skicka och ta emot meddelanden.

Abstract

When a family member is ill, there is a need to know about his wellbeing. In such cases standard messages such as “How are you feeling right now”, “I am ok now”, “I am very ill”, “Call me now”, “I am better now” need to be exchanged. Along with this, there is also a need to have a communication link which is silent, as sometimes an ill person may be sleeping. Messages could also be exchanged between friends or relatives where we may have messages like “How are you?”, “Do you want to go out”, “I am bored”, “I am leaving now” etc. In such scenarios, there is a need for quick exchange of short messages. A standard set of messages is often repeated in a conversation.

The goal with this thesis is to develop software for wireless sensors which could be used to quickly send and receive standard messages. In order to make the device mobile, Wireless Sensors were used to form a Wireless Link. Along with this a feasibility study is conducted in order to determine the user interface which could be used along with the wireless sensors.

Unfortunately, communication between two wireless sensors units generally requires a third sensor to govern the Wireless Sensor Network. This third sensor, often called “node 0” or “base station” needs to be connected to a computer. The main aim here is to find a method of eliminating the base station and enable two sensors to communicate with each other, independent of base station. Thus, the possible infrastructure required for the network is reduced.

A program is developed which enables two or more sensors to communicate with each other without the base station. After developing the program, extensive tests were carried out in order to determine the functionality of the program, the possible range of transmission and the possibility of having more than two sensors in a network.

Along with this, a mockup of the handheld device which could be used in such a network is made and interviews were conducted with professionals to determine the usability of such a device and to determine the possible user interfaces that could be included in such a device.

A conclusion that is drawn here is that, it is possible to eliminate base station from a Wireless Sensor Network and to have two or more sensors that can communicate with each other. Also, it is possible to build a handheld device based on the mockup, in order to have two individuals send and receive messages.

Keywords

Wireless Sensor Network
Motes
Mobile Communication
Short Messages

Acknowledgements

During these 20 weeks that I have been working on this master thesis, I have met many wonderful and helpful people and therefore I would like to take this opportunity to thank everyone.

First of all I want to thank my supervisor at the Department of Electroscience, Peter Nilsson for patiently considering my queries and suggesting excellent solutions for them. I would also like to thank him for giving me access to all the resources I needed throughout these 20 weeks. I would also like to thank my supervisor at Certec, Peter Waller, for all his ideas and comments on the project. I would also like to thank him for all the long discussions we had in the course of this project. His knowledge in the field of rehabilitation engineering really helped me put together my thesis. I would also like to thank Håkan Eftring and Arne Svensk for guiding me through this thesis and patiently reviewing my work.

I would like to mention a special word of thanks for my brother, Vikas for supporting me, not only in this thesis but through my entire life. Whatever I have achieved today I owe it all to him. Along with that I would also like to thank Sheryl and Yrsa for their wonderful company I had within the past one year.

This acknowledgement would be incomplete without thanking my parents in India who by their teachings and virtues have helped me shape my character.

Table of Contents

1 Introduction

1.1 Aims and Objectives

The aim of this thesis is to explore the possibility of using Motes [1.1], discussed in section 2, by developing software for a wireless communication link to be established among motes which enables users to send and receive text messages over a wireless link. These text messages could be simple day to day messages of well being, as well as other standard messages between two close friends or relatives. In order to enable a user to read the received or sent messages a feasibility study is also conducted to determine the user interface which could be incorporated in the handheld device.

Two users would have a handheld device each. Each handheld device would have four buttons, corresponding to four short text messages and a text display. When one of the users presses a button, a text message should instantly be displayed on the other user's handheld device.

This thesis focuses on the software development for the Motes, in order to establish a two way wireless link.

1.2 Problem Description

People communicate with each other in order to share their knowledge and experiences. Common forms of human communication include sign language, speaking, writing and gestures. Communication can be interactive, intentional, or unintentional; it can also be verbal or nonverbal. The concept of using nonverbal communication between two individuals to achieve 'Silent Communication' was the aim of the thesis conducted by Markus Bylesjö [1.2]. I would like to give a brief description of the work done by Markus Bylesjö in order to explain Silent Communication which led to the development of my thesis.

1.2.1 Previous work by Markus Byjesjö

Markus Bylesjö conducted his thesis at Certec [1.3] in spring 2005 using Motes. The aim of his thesis was to establish a wireless communication link between two individuals. Figure 1.1 explains the concept of his thesis.

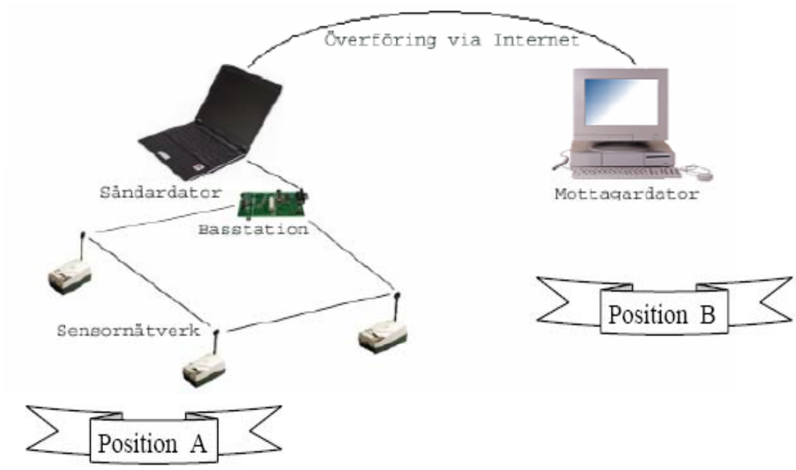


Figure 1.1 Overview of the thesis conducted by Markus Bylesjö.

As seen in figure 1.1 three MICA2 motes [1.4] were placed at position A which may be the residence of an individual or any other location. These motes were then connected to a base station (called Node 0) which was connected to a laptop computer, also placed at position A. The concept here was that these motes would send information in the form of a reading from a light or acceleration sensor to node 0. When this reading was received at node 0 the computer would use an internet connection to connect to another computer at position B to display a message corresponding to the sensor reading. When the mote at position A was covered by the user's hand a reading was sent to node 0 which in turn would flash a message on the computer at position B using TinyDB [1.5] and Windows Net meeting [1.6]. Thus, on covering the mote or by shaking it two different messages would flash on the computer at position B. In order to achieve a half duplex [1.7] communication a provision was also made to send acknowledgements using the same arrangement. On the computer at position B a software developed made use of two keys which correspond to two different messages which could be used to acknowledge the user at position A. Thus on receiving a message from position A, a person at position B could respond to it and a buzzer on the mote at position A would buzz or a combination of one or more LEDS [1.8] could be used to indicate acknowledgements or a silent question on a person's health state.

1.2.2 Problems to be solved

Thus in his work Markus Bylesjö achieved a two way communication which was silent in a way as it used sensors along with LEDS to achieve this. But along with this the setup required to have a node 0, a laptop computer at position A, an internet connection and a computer at position B. As only two messages could be sent and received from position A to B and vice versa, there was a restriction on the kind and number of messages which could be sent and received. The communication achieved was wireless which allowed the motes to be placed any where at position A, hence a user initiated a session from any place at position A but at position B a person needed to be at his desk to respond to incoming messages.

These limitations have led to the development of my thesis work which tries to eliminate some of the restrictions in the work conducted by Markus Bylesjö.

1) The main goal of my thesis is to establish a communication link which is independent of node 0, thus eliminating the laptop computer at position A. As the laptop computer at position A is eliminated we are not dependent on the computer at position B.
2) This thesis will also try to find a possible solution to extend the user interface and incorporate more than two, possibly four or more messages which could be sent and received using the motes wireless link.
3) Also another aim is to suggest a design for the handheld device as described in section 3.1, 3.2, which incorporates a LCD [1.9] display to enable the user to read messages, which he sends or receives. Figure 1.2 and 1.3 give pictorial views of the intended work. This task made use of the Universal Design [1.10] approach in which the suggested device (handheld device) would be useful in general and specific operation.
4) Developing a multi hop [1.11] network in order to increase the range of transmission.
5) Along with that, exploring the possibility of having more

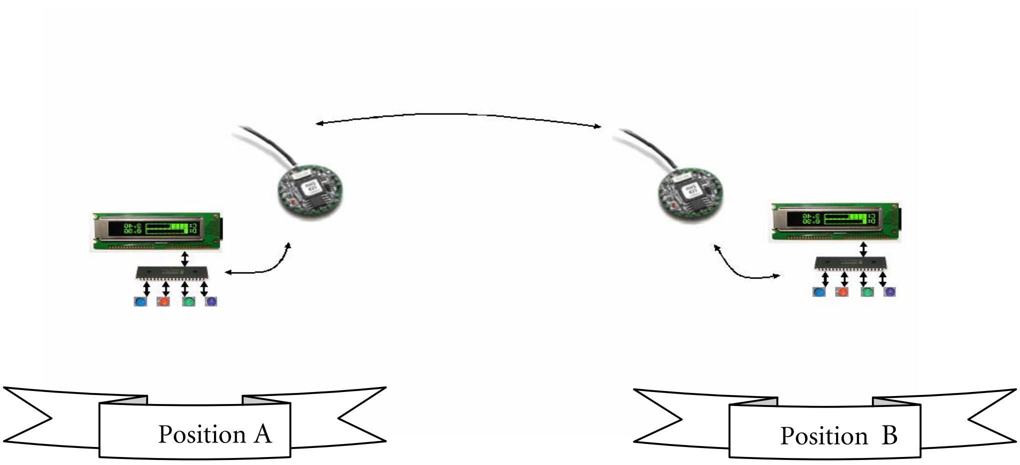


Figure 1.2 Wireless link being established between two motes without using a base station.

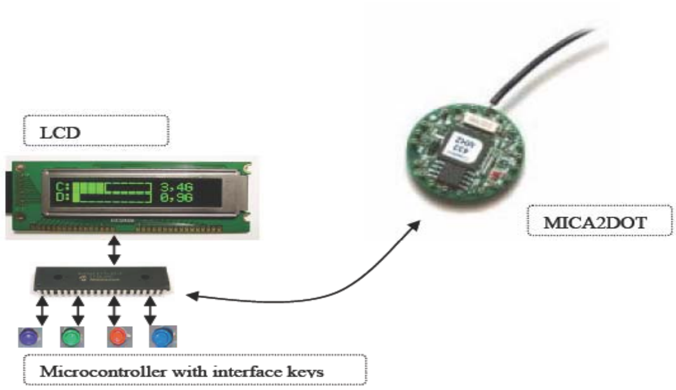


Figure 1.3 A magnified view of the proposed interface at the two positions. The MICA2DOT mote, along with its antenna, the LCD display and the microcontroller with interface keys are all suggested to be contained in one handheld device.

As the computer at both positions A and B is eliminated the communication is made mobile, in the sense that the users at position A and B could send and receive messages while being on the move. The possibility of interfacing motes with a GSM [1.12] link will also be explored to increase the range of transmission in the network.

One more aspect to be looked into is to incorporate more than two motes in an ad hoc [1.13] network thus exploring the possibility of communicating with more than two handheld devices, described in section 4.4.3.

I would like to give a real life example to explain such a scenario as illustrated in Figure 1.4. Consider a housing complex with many people. There are two individuals residing on different floors, they would like to know if the cafeteria in the basement is open or if it has closed early today. In such a case a light sensor could be placed on a third mote along with that we can have two motes (along with handheld devices) with two individuals in the network. The third mote with the light sensor would send its reading to both the individuals who then decide whether they would like to go to the cafeteria or not. In this situation we would have a half duplex communication being established.

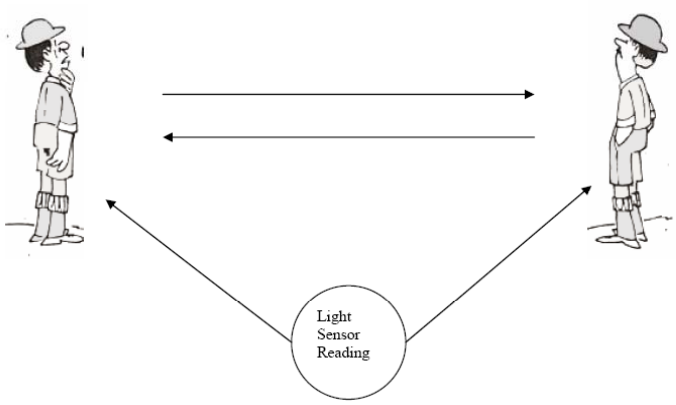


Figure 1.4 Scenario using two handheld devices and a light sensor

Another case of using three or more motes in an ad hoc network would be when we have three individuals using three motes (along with handheld devices) sending and receiving messages from each other to reach a consensus as seen in Figure 1.5. This would also require the system to achieve half duplex communication.

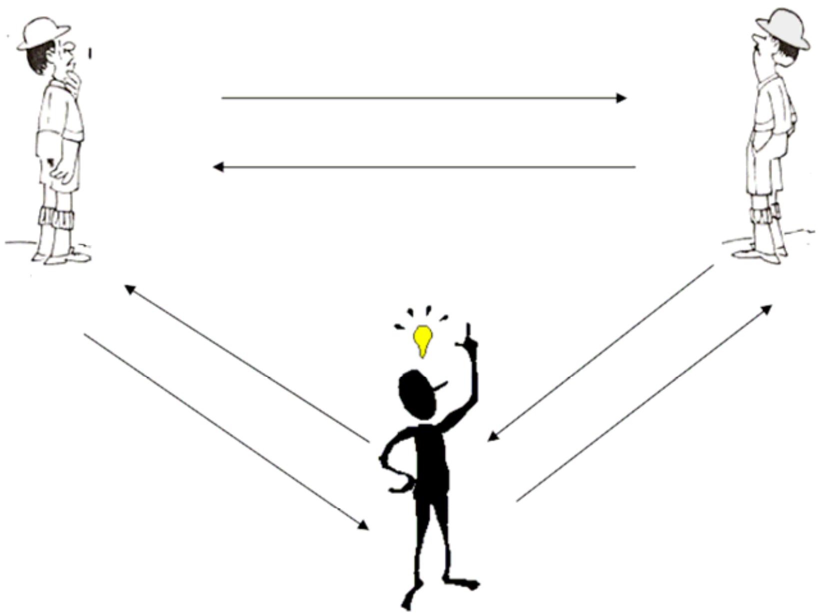


Figure 1.5 Scenario using three handheld devices in an ad hoc network

Another real life example where a simplex [1.14] communication link could be established would be a case where, Jacob a person who recently met with an accident which resulted in him having a short term memory. Every week Jacob has been asked to wash his clothes in the basement of his house, he remembers to load his clothes in the washing machine but forget to start it. Thus, his old mother has to go down to the basement and start the machine herself.

In such a case if Jacob and his mother had a communication device using short messages, then every time Jacob goes down in the basement to wash his clothes his mother could send him a message to start the machine to aid Jacob in his daily chores. In such a case this device would function as a 'reminder' thus achieving a simplex communication.

Some other scenarios of using such a communication link will be in hospitals, where patients could send short messages to their nurses regarding their comfort and needs. In such cases messages such as "I am uncomfortable", "I need water" could be incorporated in their handheld devices. In such a case the communication would be initiated by patients and the communication link could be simplex or half duplex.

1.3 Limitations

The limitations of this thesis were:

1) Motes are defined as Embedded Computers which gather, process and transmit data to a base station or a neighboring mote in an ad hoc network. Currently, the market trend is towards smaller size and low power consumption and motes fit these credentials perfectly as they have very small size. The size of a MICA2 mote is 6 x 4 cm and is powered by 2 AA batteries whereas a MICA2DOT [1.15] mote has a diameter of 25mm with a total length of 10 cm and is powered by a coin cell.

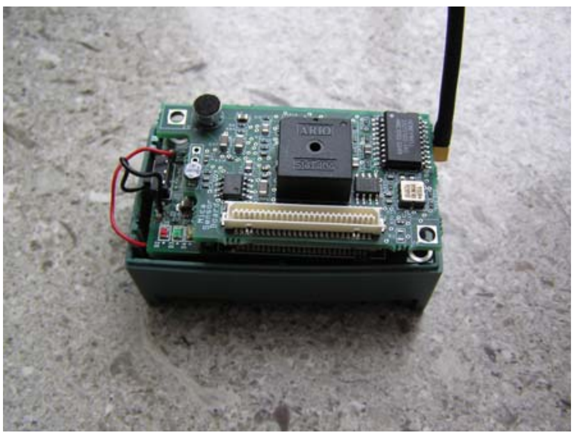


Figure 1.6 MICA2 Mote

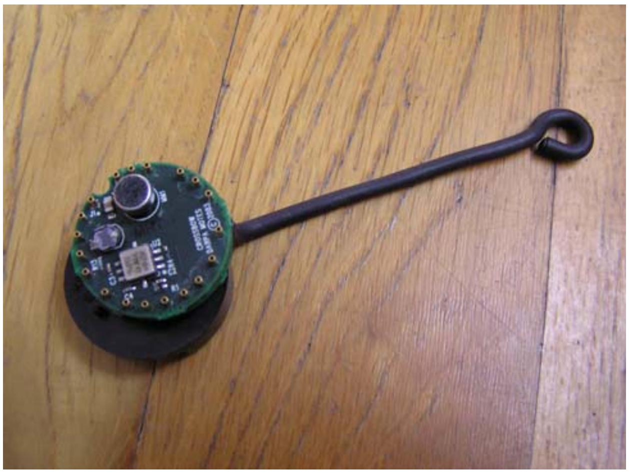


Figure 1.7 MICA2DOT mote

As this project would ultimately lead to the development of a handheld device which required to be small and light enough to be held, the smaller and lighter MICA2DOT mote is used instead of the MICA2 mote for the handheld device.

2) Increasing the range of transmission using motes is also a field in which this thesis looked into. The possibility of using GSM was one option but because of the time limitations this option was not considered but another method is devised to increase the transmission range. A program named as Forwarder described in section 4.4.4, is developed which could be installed on a number of MICA2 motes in a network to increase the range of transmission, instead of the GSM link. This program made use of the multi hop principle to increase transmission range. Figure 1.8 helps in explaining this concept.

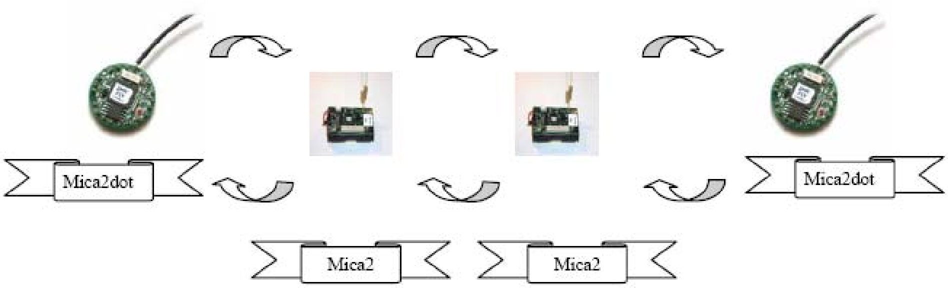


Figure 1.8 Multi hop using Forwarder installed on MICA2 motes

3) This thesis mainly looks into the possibility of using two motes to transmit and receive information in an ad hoc network. Also this aspect could be extended to having more motes in the network which would communicate without interference, for

which three motes were tested in a network in order to check for interference and usability, described in section 4.4.3.

4) In order to determine the kind of interfaces which could be used along with the software developed for the motes, a mockup of the communication device is made to gain feedback on the interface devices which could be used.

The main focus is on developing software for the MICA2DOT motes and not on the software for the microcontroller controlling the user interface. Only a mockup of the hardware is developed. However some possible hardware components were also looked into.

1.4 Method

As it is crucial to know how motes work and how they communicate with each other, a background study of the motes and their possible applications is made. Also a good deal of time is devoted in reviewing the kind of work done by Markus Bylesjö as this thesis is designed to enhance some of the aspects in his work. As the major aspect of this thesis is to eliminate node 0 a feasible solution had to be sought for which would enable two motes to communicate without the overhead of node0.

After understanding the problem and the possible solution for it, a software is designed which achieves half duplex communication and also eliminate base station. After developing the software and installing it on two MICA2DOT motes a controlled test is conducted in the laboratory which demonstrates the functionality and range of such a network.

In order to have a wider range of transmission, another program "Forwarder" is developed which is installed on MICA2 motes forming a multi-hop network.

After developing and testing the software for the motes a mockup [1.16] of the handheld device which would use the motes is designed. This mockup would enable a user to experience how the hand held device would appear physically and also how it would function. It would also give users a preview of the kind of user interfaces, in terms of the display device (LCD), keys, vibrator and buzzer which may be included in it. In this phase the Universal Design approach is used which enabled me to suggest a design for the handheld device which could incorporate a wider range of people. Hence, the system could be custom made for an elderly person with a reduced vision or for a young child who has just learnt to read and write.

Also while suggesting the kind of interfaces which could be used in the handheld device, it is made sure that the design exploited the strengths of an individual using it. For example a person could use his fingers or elbow to push the keys hence the keys on the handheld device would be placed rather apart from each other to prevent accidental pressing. It is intended to have a handheld device which would be simple to use and also would enable a user to get immediate feedback. When a user would press a key a message corresponding to it should be displayed on his LCD thus enabling him to read the message which had just been sent.

After designing the mockup, interviews were conducted with Håkan Eftring [1.17] and Arne Svensk [1.18] who are experienced professional in the field of Rehabilitation Engineering, to obtain their opinion on such a network and also on the possible device which could be developed. A detail description of their comments and suggestions has been included in section 5.

2 Motes

In order to achieve wireless communication, the wireless sensors MICA2DOT from Crossbow [2.1] were used. These sensors were chosen mainly for their small size and also for their ability to communicate up to a distance of 60 meters. The intension is to design a software which could be installed on MICA2DOT motes to form a network of two or more motes, which would enable users to communicate with each others using text messages. The wireless sensors used here are micro electro mechanical sensors also known as "**motes**". A wireless sensor network is a collection of wireless sensors (motes) that form a certain network topology [2.2]. Sensor data is collected from a certain location, locally processed or aggregated and transmitted to one or more base stations or neighboring motes. A sensor node (mote) combines the abilities to compute, communicate, and sense. Thus, a sensor node consists of a processing unit, communication module (radio interface) and sensing device. Each sensor node delivers the collected data to one (or more) neighboring nodes. By following a multi-hop communication paradigm data is routed to the sink (base station or a neighboring mote) and through the base station to the users. Therefore, multi-hop ad hoc techniques constitute the basis also for wireless sensor networks.

Generally, when people consider wireless devices they think of objects such as cellular phones [2.3], personal digital assistants (PDA)[2.4], or laptops using WLANs e.g. 802.11[2.5]. These items cost hundreds of dollars and target general purpose applications. In contrast, wireless sensor networks use small, low-cost embedded devices for a wide range of applications and do not rely on any pre-existing infrastructure. Crossbow aims to have devices that would cost less than $1 by 2007.

2.1 General Concept

The "mote" concept creates a new way of thinking about sensors, but the basic idea is simple: The core of a mote is a small, low-cost, low-power microcontroller. The microcontroller connects to the outside world with a radio link. The most common radio links allow a mote to transmit at a distance of something like 60 meters. Power consumption, size and cost are barriers to longer distances. Since a fundamental concept with motes is tiny size, small and low-power radios are normally used. This microcontroller can also monitor one or more **sensors**. Such sensors could be used to study temperature, light, sound, position, acceleration, vibration, stress, weight, pressure, humidity, etc.

2.2 Different application areas for sensors

Generally, human to human communication using sensors (motes) is not very common, whereas sensors are used to interact between people and computers. An example here would be a temperature sensor which senses a person's temperature and sends the reading to a computer to be processed. [2.A1] Along with this I would like to present some of the areas in which sensors have been used recently.

2.2.1 Sensors watch Barrier Reef coral

"With the ecology of coral reefs around the globe increasingly under pressure, scientists on Australia's Great Barrier Reef are establishing a network of sensors to better understand this beautiful part of the underwater world. The Australian Institute of Marine Science (Aims) is working with James Cook University on a project called Digital Skins. Smart sensors, developed originally for use in nuclear power stations, are placed in the ocean and also in water catchments on the mainland".

2.2.1.1 Shared computing concept

"Each sensor in the skin has its own numerical address and operating system. Using a global position system, the sensors know exactly where they are and parameters such as salinity, temperature and nutrient levels are measured. Communicating with sensors is a challenge, particularly for those sensors located out on the reef. Using a technique that was discovered by the British during World War II, in which microwave signals are sent along the surface of the ocean. Initial tests have seen data sent as far as 70km (43.5 miles) in one hop. The final link in the chain is grid computing, all these sensors create terabytes of data every day and high-speed links allow the various institutions to share their computing power" [2.A2].

2.2.2 Elephant seals dive for science

Figure 2.1 The unit is stuck down and will fall off when the animal moults

“Elephant seals on South Georgia have been recruited to the cause of science. Equipped with computerized tags stuck onto their heads, the animals have been collecting remarkable new information on conditions in the Southern Ocean. As the animals swim for thousands of km and dive down to 2,000m, their tags record details of temperature, depth and the salinity of the water. When the seals pop up to breathe, the computers transmit the information to scientists in Scotland via satellite.
The elephant seals' information has enabled researchers to study how changes in salinity and temperature affect the movement of water at different depths. This has provided new insights into the habits and habitat both of the elephant seal and its prey species, squid. However, it has also significantly improved in understanding of the processes of heat exchange within the Southern Ocean and between this region and the rest of the world”. [2.A3]

2.2.3 Green light for tsunami sensors

“Indian Ocean countries meeting in Australia have decided to set up a network of seabed sensors and buoys as part of a tsunami warning system. The sensors are sensitive enough to detect the slightest changes in water column pressure. They will transmit information to sea-level buoys. The meeting in Perth was attended by representatives of 27 Indian Ocean countries at risk from tsunamis. Tsunami in December 2004 has killed more than 220,000 people. The warning system could be fully operational by summer 2006. The establishment of seismic and sea-level networks for the Indian Ocean and an agreement on how to deploy those was believed to be a significant step forward by Neville Smith, vicechairman of Unesco's Intergovernmental Oceanographic Commission (IOC)". [2.A4]

2.3 MICA2DOT Mote

2.3.1 Some features of MICA2DOT

MICA2DOT is the third generation, quarter-sized (25mm in diameter), wireless platform for Wireless Sensors. They are designed specifically for deeply Embedded Wireless Sensor Networks [2.6]. They are battery-powered and light weight as a result they are preferred for a number of applications, monitoring of wildlife being one such application. They posses routing capabilities, hence they are used to forward information to neighboring nodes. They operate at 868/916 MHz, 433 MHz or 315 MHz multi-channel radio transceiver and are compatible with the larger MICA2 motes. MICA2DOT makes use of TinyOS (TOS) [2.7] Distributed Software Operating System v1.0 with improved networking stack and improved debugging feature.

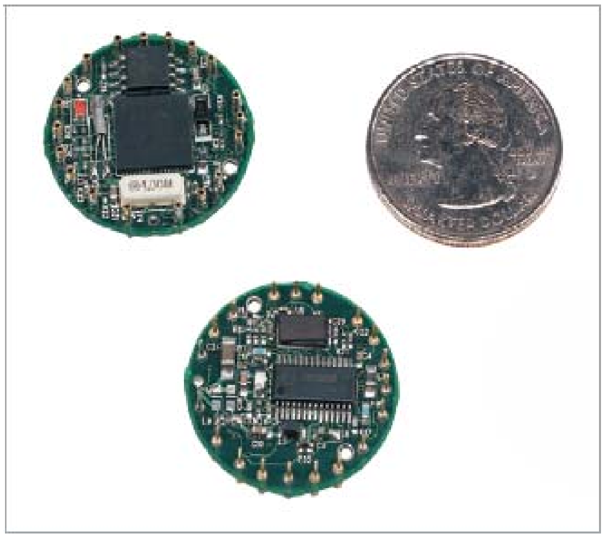


Figure 2.2 MICA2DOT motes [2.A5]

2.3.1.1 Processor and RadioPlatform:

The MICA2DOT is based on the Atmel [2.8] ATmega128L [2.9] The ATmega 128L is a low power microcontroller which runs TOS from its internal flash memory. Using TOS, a single processor board (MPR500CA), as seen Figure 2.3 [2.10] can be configured to run sensor application/processes and the network/ radio communications stack simultaneously. The MICA2DOT features 18 solder less expansion pins for connecting analog inputs, digital

I/Os and a serial communication or UART interface [2.11]. These interfaces make it easy to connect to a wide variety of external peripherals.

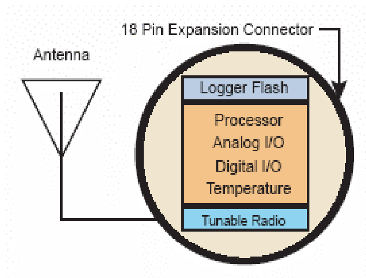
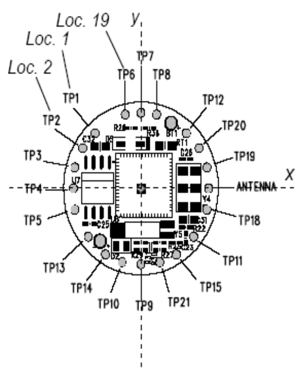


Figure 2.3 MPR500CA Block Diagram [A5]

2.3.1.2 MICA2DOT Expansion Connector

MICA2DOT consists of a set of control pins, ADC (Analog to Digital Converter) ports, power pins, ground, some general purpose digital and analog I/Os and the serial programming port (UART).



Loc.	x	y	Pin	Name	Description
1	-0.290	0.315	TP1	GND	Ground
2	-0.370	0.230	TP2	ADC7	ADC Channel 7
3	-0.420	0.120	TP3	ADC6	ADC Channel 6
4	-0.430	0.000	TP4	ADC5	ADC Channel 5
5	-0.420	-0.120	TP5	ADC4	ADC Channel 4
6	-0.335	-0.275	TP13	ADC3	ADC Channel 3
7	-0.225	-0.375	TP14	ADC2	ADC Channel 2
8	-0.120	-0.420	TP10	UART_RX	UART Receive
9	0.000	-0.430	TP9	UART_TX	UART Transmit
10	0.120	-0.420	TP21	THERM_PWR	GPIO
11	0.225	-0.375	TP15	PWM1B	GPIO
12	0.335	-0.275	TP11	RESETN	MCU Reset
13	0.420	-0.120	TP18	GND	Ground (Battery –term)
14	0.420	0.120	TP19	INT0	GPIO
15	0.370	0.230	TP20	INT1	GPIO
16	0.290	0.315	TP12	SPI_CLK	Radio Clock
17	0.100	0.420	TP8	PW0	GPIO/PWM
18	0.000	0.430	TP7	PW1	GPIO/PWM
19	-0.100	0.420	TP6	VCC	Voltage (Battery +term)

Figure 2.4 MICA2DOT expansion connector [1.15A]

2.3.2 TinyOS and nesC

TinyOS is an open-source operating system designed for wireless embedded sensor networks. The programs running on TinyOS are written in nesC [2.12], a language with a C like syntax. It features a component based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS's component library includes network protocol, distributed services, sensor drivers and data acquisition tools-all of which can be used as it is or be further refined for a custom application. TinyOS's event-driven execution model enables fine-grained power management yet allows the scheduling flexibility made necessary by the unpredictable nature of wireless communication and physical world interfaces. [2.A7]

2.3.3 TinyOS Programming Philosophy

In order to design the software for the motes it was important to understand the TinyOS programming philosophy. The TinyOS operating system, libraries, and applications are all written in nesC, a new structured component-based language. The nesC language was constructed to be modular, allowing the user to snap together pieces of code and is primarily intended for Embedded Systems. nesC has a C-like syntax, but supports the TinyOS concurrency model[2.13], as well as mechanisms for structuring, naming and linking together software components into robust embedded systems. The principle goal is to allow application designers to build components that can be easily composed into complete and concurrent systems. [2.A7]

2.3.4 TinyOS Concurrency Model

TinyOS executes only one program consisting of selected system components and custom components needed for a single application. There are two threads of execution: tasks and hardware event handlers. Tasks are functions whose execution is deferred; once scheduled they run to completion and do not pre-empt one another. Hardware event handlers are executed in response to a hardware interrupt and also run to completion. Unlike a task, it may pre-empt the execution of a task or other hardware event handler

As mentioned previously, the primary building blocks of nesC code are components. These components are of two types, modules and configuration.

Module: A module is a component that implements one or more interfaces.
Configuration: A configuration is a component that wires other components together; connecting interfaces used by the components to interfaces provided by others.
Interface: An interface is used to provide an abstract definition of the interaction of two components. It is the only means of data transfer in a component. They link commands and events

provided by one component to another that wishes to use them. The interface for a component is defined in a module. The commands, events and task passed by interfaces are defined in implementations.
Commands: Commands are set of functions that the interface provider must implement.
Events: Events are another set of functions the interface provider must implement.
Application: An application is run-time executable that is created by linking / wiring together one or more components.

The Figure 2.5 represents a typical nesC application that references standard system files and contains some of its code.

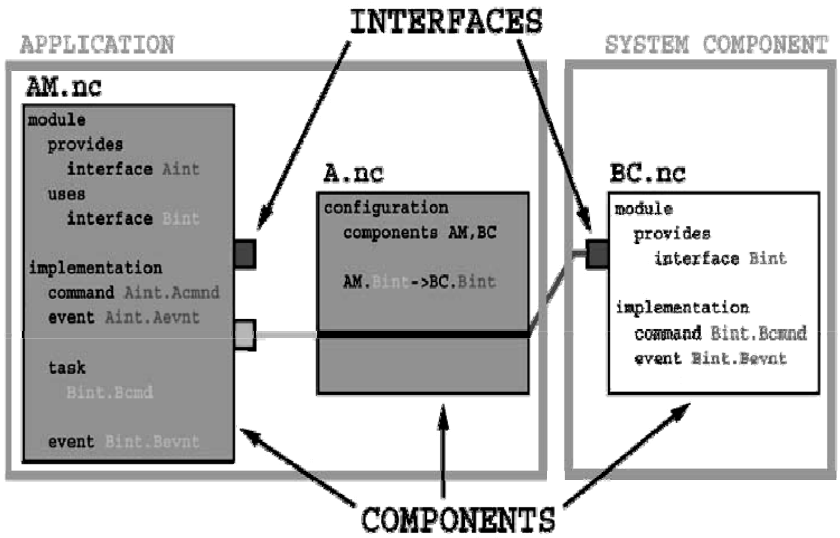


Figure 2.5 A typical nesC application

From figure 2.5, module BC.nc *provides* the interface Bint. The implementation for the module defines a command Bint.Bcmnd and an event Bint.Bevnt. E.g. if we were programming module AM.nc and wanted to use the command Bint.Bcmnd, in order to gain access to this command, in the declaration of our module we would say it *uses* the interface Bint. We would now write a separate configuration file A.nc which wires the interface of module AM.nc to the interface of module BC.nc. Module AM.nc now has access to all of the commands provided by BC.nc. A typical implementation for module AM.nc might define commands or events provided by AM.nc. The implementation might also contain a *task* where it uses the command from BC.nc. However, a fundamental rule of nesC states that any module that uses a command from another module must address all of the events provided by the other module. Hence, AM.nc must also contain code to address the event Bint.Bevnt.

2.3.5 Cygwin

Cygwin, seen in figure 2.6, is a collection of free software tools originally developed by Cygnus Solutions [2.14] to allow various versions of Microsoft Windows to act somewhat like a UNIX/Linux system. It aims mainly at compiling software that runs on POSIX systems [2.15] (such as GNU/Linux systems, BSD systems, and UNIX systems) to run on Windows with little more than a recompilation. It consists of two parts:

A DLL (cygwin1.dll) which acts as a Linux [2.16] API [2.17] emulation layer providing substantial Linux API functionality. The Cygwin DLL works with all non-beta, non "release candidate", ix86 32 bit versions of Windows since Windows 95, with the exception of Windows ME. For this project it is basically used to embed software code into the motes, with the help of MIB 510 programming board.

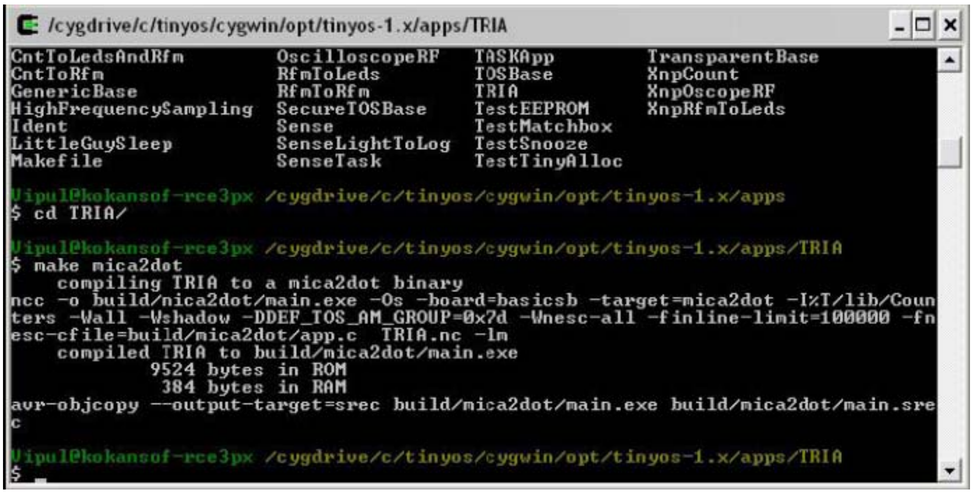


Figure 2.6 Cygwin screenshot while compiling TRIA [Ref. section 4.1]

3 Hardware overview

As design of the handheld device is not the main aim of this thesis, a feasibility study is conducted in terms of the physical appearance of the device and also to determine the kind of user interfaces which could be included in it. A mockup of the proposed handheld device helped me in collecting feedback in terms of its functionality, size and appearance. In this section I would like to describe the possible hardware for the handheld device and the mockup created.

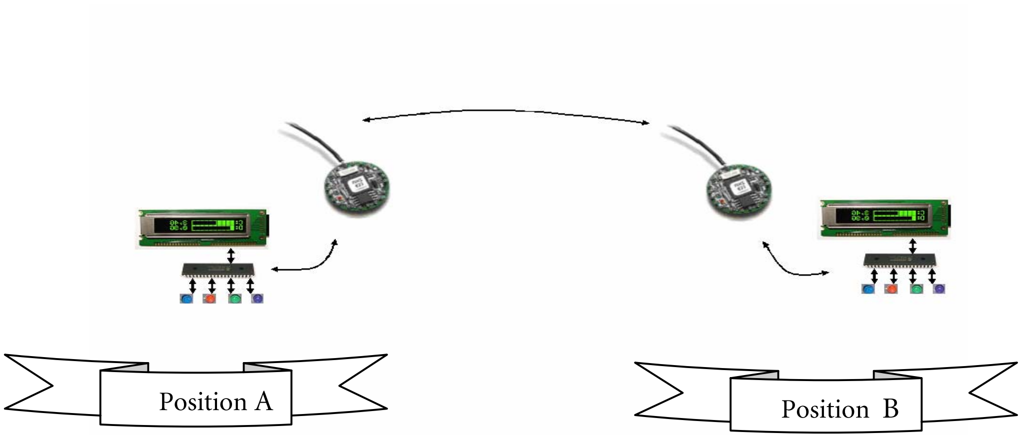


Figure 3.1 Wireless link being established between two motes without using a base station.

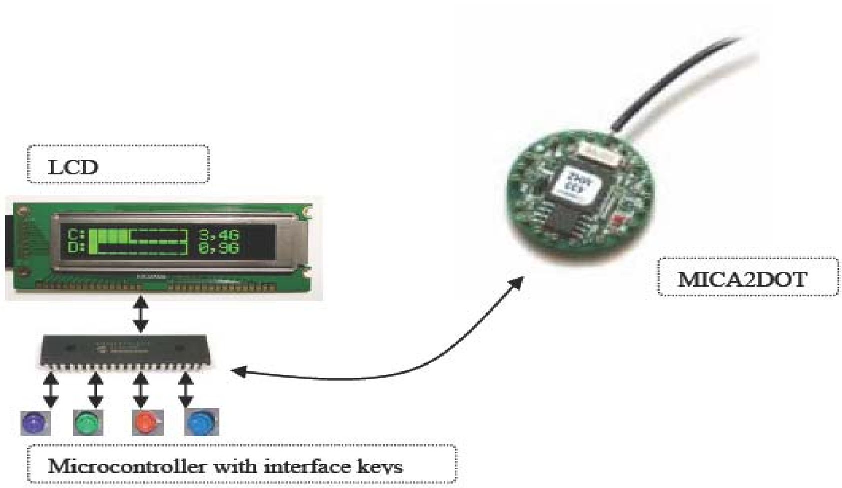


Figure 3.2 A magnified view of the proposed interface at the two positions. The MICA2DOT mote, along with its antenna, the LCD display and the microcontroller with interface keys are all suggested to be contained in one handheld device.

Figure 3.1 describes the wireless link being established between position A and B in order to achieve a two way communication for sending and receiving text messages. Figure 3.2 provides a magnified view of the intended user interface. The concept here is that each key on the handheld device corresponds to a different message and when one key (at position A) is pressed a message corresponding to it would be displayed on the LCD at position A and also on the LCD at position B. On receiving a message from a person at position A an individual at position B could respond to the incoming message by using some of his preprogrammed keys, thus achieving a two way communication.

3.1 Mockup design for intended hand held device

It is intended that the interface devices depicted in the figure 3.2 should be placed in the handheld device. Hence the mockup created included a LCD; four color coded keys and a MICA2DOT mote. As the intended handheld device would have a battery supply along with a microcontroller it is made sure that the mockup represented the handheld device not only in functionality but also in its weight and dimensions. The mockup had size of 13cm x 7cm with a weight of 110 grams.

Figure 3.3 Mockup of the handheld device with four keys and four messages in both directions.

As seen in the figure 3.3 the mockup had a LCD display which would display incoming and outgoing messages. The mockup also consisted of four keys of different colors which would enable a user to differentiate between each key and the message it corresponds to. As this mockup made use of the Universal Design approach, it is made sure that the design exploited the strengths of an individual using it. For example a person could use his fingers or elbow to push the keys hence the keys on the handheld device would be placed rather apart from each other to prevent accidental pressing. It is intended to have a handheld device which was simple to use and also would enable a user to get immediate feedback. When a user would press a key, a message corresponding to it would be displayed on his LCD thus enabling him to read the message which had just been sent. This made sure that the right message, intended by the user has been sent to a neighboring handheld device.

3.2 Suggested Hardware for the handheld device

In order to design a handheld communication device a number of factors needed to be considered. The device needed to have a display which would enable users to read the messages they send and receive. Along with that a few keys were needed to be placed on the device which would be preprogrammed to display a specific message on the display. A standard LCD (Liquid Crystal Display) screen could have been used for the device but as the device is made using the Universal Design approach, a display device that would be suitable for general and specific applications had to be sought for. The requirements of the LCD device were that the size of the characters should be about 8mm and also it should be bright enough for a person to read in daylight or in the dark. A standard LCD would not be able to meet all these requirements hence an OLED LCD see section 3.2.1[3.1] (with 20x2 lines using green color display and text height of 8mm) was suggested as a possible alternative.

3.2.1 OLED:

OLED see figure 3.4, short for organic light-emitting diode is a display device that sandwiches carbon-based films between two charged electrodes, on a metallic cathode and a transparent anode, usually being glass. The organic films consist of a hole-injection layer, a hole-transport layer, an emissive layer and an electron-transport layer. When voltage is applied to the OLED cell, the injected positive and negative charges recombine in the emissive layer and create electro luminescent light.

Figure 3.4 An OLED LCD screen

Unlike LCDs, which require backlighting, OLED displays are emissive devices - they emit light rather than modulate transmitted or reflected light. OLED technology was invented by Eastman Kodak in the early 1980s. It is beginning to replace LCD technology in handheld devices such as PDAs and cellular phones because the technology is brighter, thinner, faster and lighter than LCDs, use less power, offer higher contrast and are cheaper to manufacture.

Along with having a luminant display device it was also necessary to accommodate a number of keys to be placed on the handheld device. These keys needed to be color coded and each would correspond to a specific message which would be displayed on the LCD. Also the keys were needed to be fairly large in size and also placed apart from each other. In order that a user knows that he has a message a buzzer along with a vibrator could also be included in the device to indicate incoming messages.

3.2.2 Microcontroller:

An OLED LCD, a buzzer, a vibrator and four keys could be interfaced with a microcontroller. For such an application any microcontroller such as PIC 16F877A or PIC 16F874 could be used mainly because of their flexible operating voltage (from 2 to 5.5V) and also for the surplus number of pins which would be available even after interfacing the above mentioned devices. Also if more keys were to be added or some additional interface device needed to be included in a custom made application then this could be achieved without having to change the microcontroller.

3.3 Software Description for the suggested handheld device

The software for the microcontroller could be made event driven, thus when a key connected to an input pin of the microcontroller is pressed a message corresponding to it would be displayed on the LCD.

3.3.1 Program Concepts while Transmitting

When a user presses a key, a message corresponding to that key would be displayed on his LCD screen and simultaneously the message would be displayed on the LCD of another user with a similar device see Figure 3.1 and 3.2.

Hence, in order to have wireless communication, a voltage controlled output from the microcontroller of the handheld device would to be given to the ADC pin see figure 3.4A of a MICA2DOT mote. This can be achieved by setting four pins (one for each key) on the microcontroller of the handheld device as output and assigning a logic 1 (3 or 5 volts) to each of them when a corresponding key was pressed or 0 when not pressed. Then, by making use of the voltage divider rule [3.1A] and using a combination of resistances, four different voltages can be obtained at each output pin. Thus, when each key is pressed a different voltage would be obtained at the output pin of the microcontroller of the handheld device. This voltage can then be given to the ADC pin of the MICA2DOT so that it could be sent over the RF link.

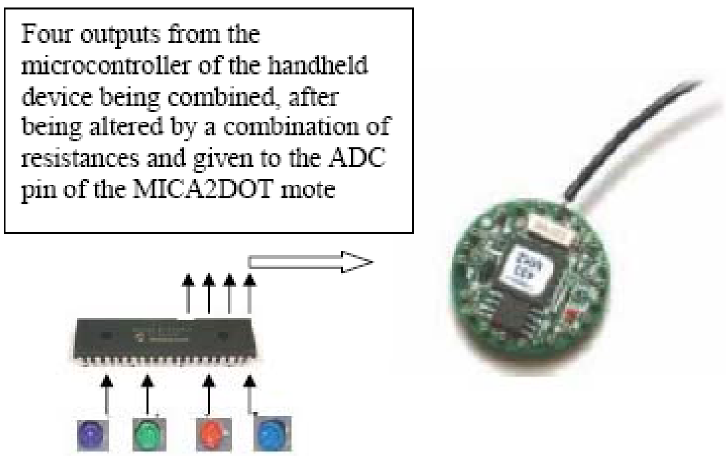


Figure 3.4A Voltage corresponding to each key being given to the ADC pin of the MICA2DOT mote.

In order to make sure that the pin remains high for a specific period of time a timer in the microcontroller could be used to count for 20 seconds each time a key is pressed such that the pin remained high for 20 seconds and then turned low. Hence, when the pins are high the message can be read at the receiver and the user at the receiving end can respond to the incoming message after 20 seconds. This window of 20 seconds could be changed after the user becomes more accustomed to the device. This method of using a timer can be considered over the Push to Transmit (PTT) method in which the pins can be made high only when the key is pressed else they is made low. The PTT method is a faster way of communicating but there is a smaller time window in which the message had to be read and understood.

3.3.2 Program Concepts while Receiving

When the program is in the receiving mode three pins can be set as input in the microcontroller of the handheld device. When a specific combination of pins goes high or low a message corresponding to that combination can be displayed on the receiving user's handheld device see figure 3.4A.

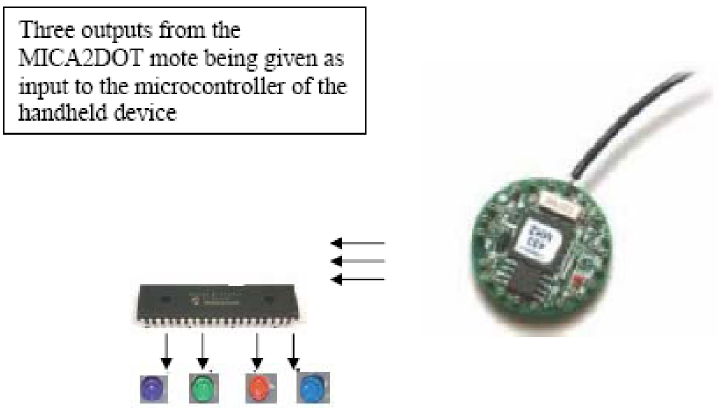


Figure 3.4B Output from the I/O ports of the MICA2DOT mote being given as input to the microcontroller of the handheld device.

Consider an example; the message transmitted from position A to B is "I am Ok!" This message corresponds to key1 and the corresponding output voltage is 0.5V. This voltage is given to the

ADC pin of MICA2DOT mote at position A which transmits it over the RF link using the developed software called TRIA, see section 4.1 for details, which has been installed on the MICA2DOT mote. At the receiver, corresponding to 0.5V the I/O port1 can be made high, also making use of TRIA which is installed on a mote at the receiver. The microcontroller of the handheld device at the receiver should be configured to display the message "How are you!", when this combination (1, 0, 0 for I/Os (1 to 3)) exits. Thus the message is sent from a user 1 to user 2 and vice versa. Along with displaying the message the program could also stimulate a buzzer and vibrator to indicate an incoming message.

3.4 Device usability and error handling

The intended device could be designed such that the person who is sending the message is aware of what he is sending. In order to achieve this, the message being sent could also be displayed on his LCD. If in case, a person presses a wrong key, he can rectify his mistake by pressing the right key at once. For example, if a user presses "I am Ok!" instead of "How are you?" He can at once press the key corresponding to "How are you?" to correct his mistake. Also the program on the microcontroller of the handheld device can be made such that no pins are turned high when two or more pins are pressed simultaneously.

3.4.1 Customizing the user interface

The handheld device could also be custom made as to have a user specific set of messages programmed on a number of keys. Along with that the users name could also be included in the message, so that the receiver knows as to who has sent the message. This would be similar to a SMS [3.2] in mobile phones which displays the sender's phone number. For example, the message "I am Ok! Erik" indicates that it was Erik who has sent this message.

3.4.2 Possible upgrades

Depending on the kind of the application more keys and thus more messages could be included in a handheld device. Along with this some other interface devices like a speaker could also be included in a custom made application. The program TRIA designed for the MICA2DOT mote has been designed to incorporate seven different messages while transmitting and receiving, described in section 4.1.

4 Software design for motes

The software to be designed for the wireless link between the handheld devices had to achieve half duplex communication. The software needed to be two fold, that is, it should be able to perform both the transmission and receiving function. The scheme is to develop a software which could be installed on a number of motes as to form a network of motes. As mentioned in section 3.3 a voltage controlled output would be obtained on an output pin of the microcontroller. Thus every key on the user interface would correspond to a different output voltage. Hence, the software developed for the MICA2DOT motes will have to differentiate between these voltages and correspondingly transmit a unique 16 bit value using the motes RF link [4.1]. The program that is developed for this purpose has been named TRIA (Transmit receive using I/Os and ADC).

4.1 TRIA

As the amount of hardware resources in MICA2DOT motes is limited in terms of the number of pins see section 2.3, software for a MICA2DOT mote had to be designed with the minimum hardware requirements. Having this setting in mind, a program that would use minimum hardware requirements had to be developed. Also a major requirement of this program developed is to eliminate the base station from the ad hoc network, which would enable two or more motes to communicate with each without the overhead of node0. It was thought of having a program which would make use of only I/O pins of the MICA2DOT mote to accept and also transmit information over the RF link. However, the limited number of I/O pins on the MICA2DOT and also taking into consideration the possible upgrade of the user interface with more keys made it a restricted solution, see section 3.4.2. In order to have a flexible solution which could be upgraded in terms of the number of keys used and which would also not occupy any additional pins on the MICA2DOT mote, the TRIA program is developed. As a voltage control output was obtained from the microcontroller the ideal way to send that voltage over the RF link was to give it to an ADC (Analog to Digital Converter) pin. The program designed required the ADC pin to be set as input which is achieved by using the

hardware.h
locatedin\cygwin\opt\tinyos1.x\tos\platform\MICA2DOT.In TRIA the ADC pin which received the input voltage converted it to a 16 bit value which is then transmitted over the RF link. When these bits were received by a neighboring mote (which also had the same program TRIA) on a handheld device, the program at the receiver mote would make one or more I/Os high or low depending on the kind of the input which is given to the ADC pin of the transmitting mote and vice versa.

4.2 Program Description

A brief program description has been presented in this section. Along with this a description of the files created and the detail flowchart of the program has been included in Appendix C.

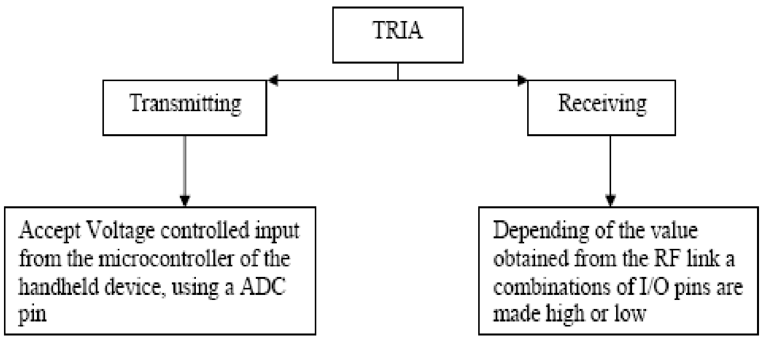


Figure 4.1 An overview of TRIA

As seen from figure 4.1, the TRIA program had two separate functions while transmitting and receiving. When in transmitting mode, it accepted a voltage controlled input (using its ADC pin) from the microcontroller of the handheld device and transmitted a unique 16 bit value over the RF link. While in receiving mode it would make a combination of I/O pins high or low, depending on the value obtained from the RF link.

4.3 Ad Hoc Network

This program developed would form an Ad Hoc network which would enable two MICA2DOT motes to communicate with each other, without making use of the Base station. Most sensors make use of a base station where data collected from remote motes is processed and as this program did not make use of node0, it is quite unique.

Also as node 0 is eliminated from the network, such a network could be set up to be independent of the existing infrastructure such as computers and internet connection.

4.4 Software tests

4.4.1 Test 1 : Operation testing

4.4.1 Overview:

A test is setup with the intension of testing out the program which is designed for the MICA2DOT motes. The program is designed to accept a voltage controlled input at position A and correspondingly make I/O pins high or low at the receiver at position B and vice versa, see figure 4.2.

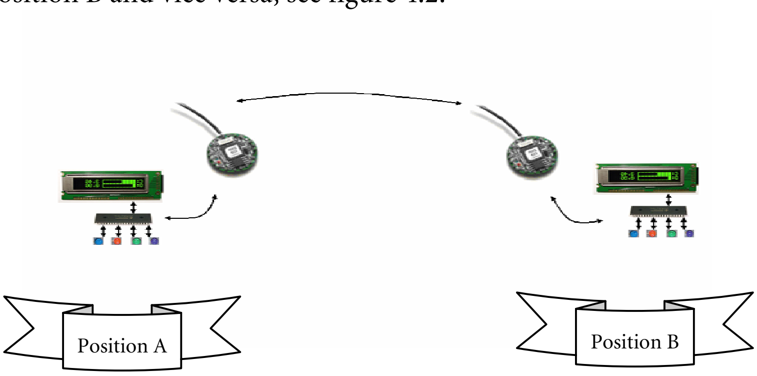


Figure 4.2 Wireless link being established between two motes without using a node0.

4.4.2 Procedure:

As described in the section 3.1, 3.2 and 3.3 the handheld device would have pre-programmed buttons which would correspond to a specific message and also a specific voltage. When a key is pressed a message corresponding to it would be displayed on the LCD and simultaneously a specific voltage (varying from 0 to 3V) would be given as input to the ADC pin of the MICA2DOT mote. In order to simulate an input voltage from 0 to 3V a variable voltage supply is used, as seen in Figure 4.4. The circuit for the MICA2DOT mote along with the variable power supply has been described in the figure 4.3A and 4.3B. Two circuits like the one shown in figure 4.3 were used to represent two handheld devices each at position A and position B.

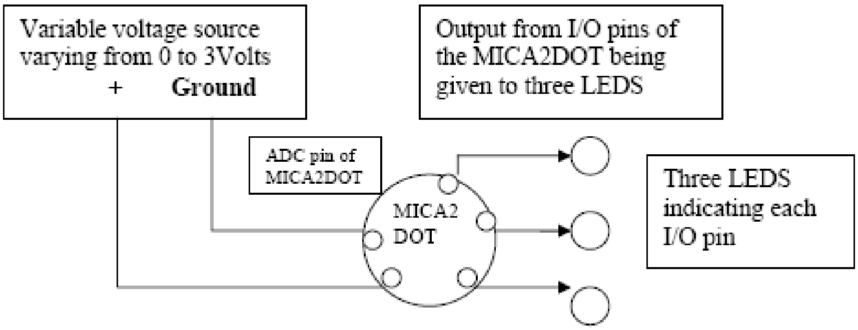


Figure 4.3A Circuit used for testing TRIA

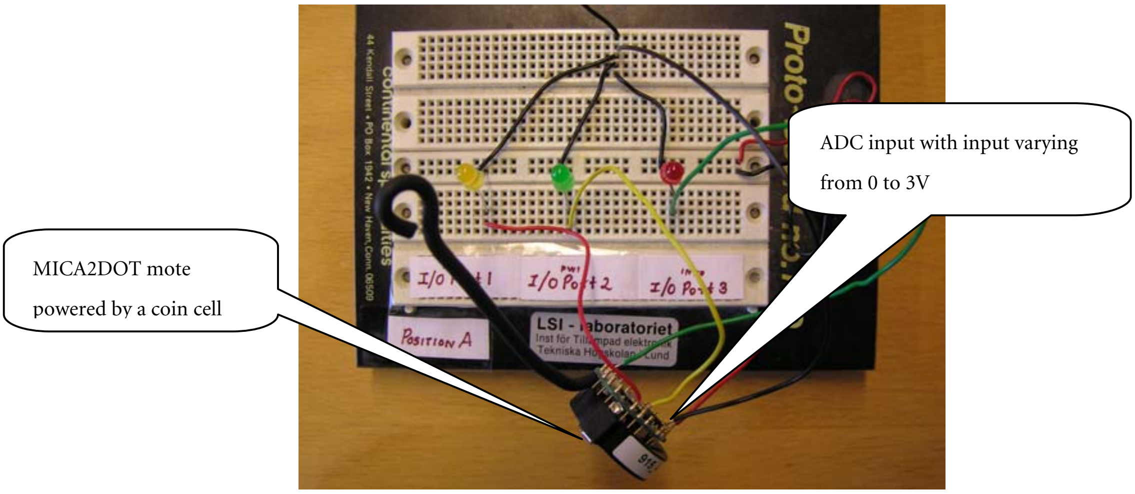


Figure 4.3B A MICA2DOT mote with ADC and I/O ports

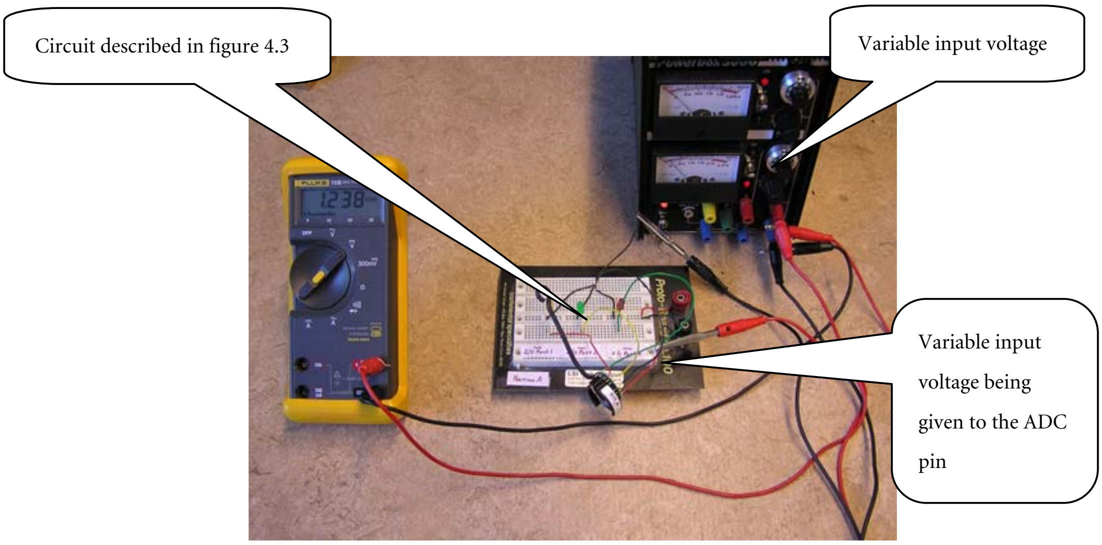


Figure 4.4 Setup used at the two positions, A and B

In order to simplify the test, a circuit described in the figure 4.3 is designed, which made use of LEDs to depict the I/O ports on the MICA2DOT mote. When an input (varying from 0 to 3 V) is given as input to the ADC6 pin of a MICA2DOT mote at Position A, then depending on the value of the input being given, a different combination of I/O ports can be made high or low at the Position B and vice versa.

Figure 4.5 would be useful in describing this process .

Input at Position A	Corresponding output at Position B		
Volts measured	I/O Port 1	I/O Port 2	I/O Port 3
0.3 to 0.6	✓	×	×
0.63 to 0.84	×	✓	×
0.9 to 1.12	✓	✓	×
1.14 to 1.4	×	×	✓
1.49 to 1.7	✓	×	✓
1.77 to 2.0	×	✓	✓
2.2 to 3.0	✓	✓	✓

Figure 4.5 Possible inputs at position A and the corresponding outputs at position B

In this case three ports on the MICA2DOT mote at position B have been configured as output namely PW0 (I/O Port1), PW1 (I/O Port2), INT0 (I/O Port3).
Similarly the results obtained when the input is given to the ADC pin of MICA2DOT mote at position B and the corresponding effect it would have on the MICA2DOT mote at position A have been depicted in the figure 4.6.

Input at Position **B**	Corresponding output at Position A		
Volts	I/O Port 1	I/O Port 2	I/O Port 3
0.3 to 0.6	✓	×	×
0.62 to 0.82	×	✓	×
0.84 to 1.12	✓	✓	×
1.13 to 1.4	×	×	✓
1.42 to 1.7	✓	×	✓
1.74 to 1.98	×	✓	✓
2.1 to 3.0	✓	✓	✓

Figure 4.6 Possible inputs at position B and the corresponding outputs at position A

In this case three ports on the MICA2DOT mote at position A have also been configured as output namely PW0 (I/O Port1), PW1 (I/O Port2), INT0 (I/O Port3)
Thus we can conclude that the program developed for the mote performed in both scenarios of transmitting and receiving from position A to position B and vice versa thus achieving half duplex communication. Also from the tests it is seen that seven different combinations were obtained thus enabling us to have seven different keys on each user interface.

4.4.3 Test 2: More than two motes in a network

Another test is conducted to explore the scenario described in the section 1.2, figure 4.7. In such a case three handheld devices would be placed in a network to send and receive text messages. In order to simulate such a situation three circuits like the one described in figure 4.3 and 4.4 were used.

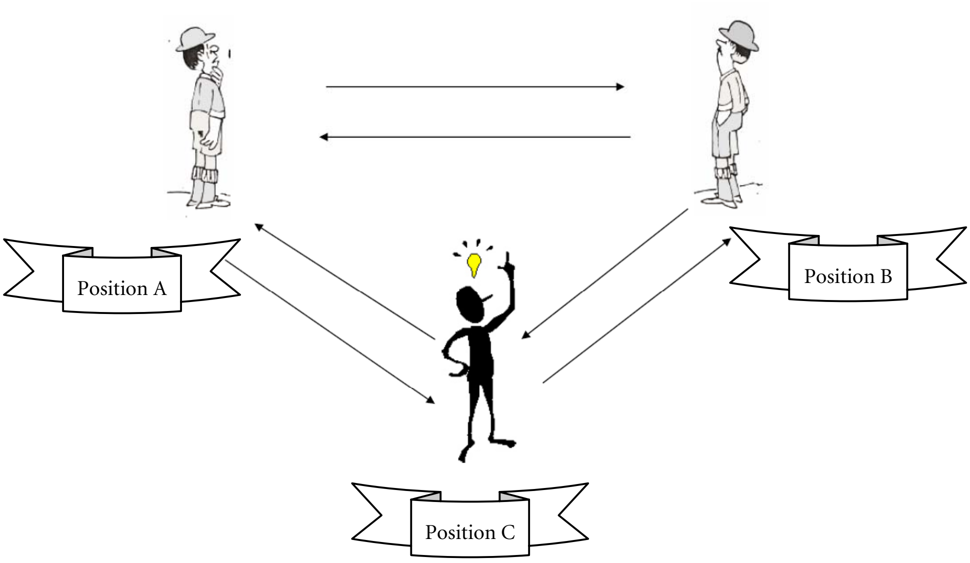


Figure 4.7 Scenario using three handheld devices in an ad hoc network

Here I would like to use a sample message and a sample voltage corresponding to it to explain such a case. Let us consider a message "I am ok!" corresponds to key 1 on the handheld device of a user at position A see Figure 3.3. Say that this message needs to be read by the user at position B. Let key 1 on the handheld device at position A correspond to 0.5 volts thus the expected output at position B and also C would be 1-0-0 on the I/O ports see figure 4.4 and 4.5. Now if the microcontroller on the handheld device at position B has been configured to display a message "I am ok!" when the combination 1-0-0 is true, only then the message would be displayed on the LCD screen at position B. Thus a message from position A has been read at position B. While this message is being send from A to B a handheld device at C will also accept the input from A. Thus the microcontroller at C also gets the input 1-0-0 from the I/O ports of its MICA2DOT mote. However, if the microcontroller at C has also been configured to display the message "I am ok!" when the combination 1-0-0 is true then the message would also be displayed on the LCD at position C else no message would be displayed. Thus while designing such a network for a number of motes there would interference in the network, that is a reading from one mote will be send to all the neighboring motes. However, while programming the microcontrollers on the handheld device it has to be made sure as to which message corresponds to which combination of I/O ports on the MICA2DOT mote.

4.4.4 Test3: Possible range of transmission

Another test is conducted to determine the maximum possible range of transmission. For this test two circuits described in figure 4.3 and 4.4 were used. This test made use of the fact that there would not always be a Line of sight (LOS) [4.2] between transmitter and receiver at position A and B see figure 4.8.

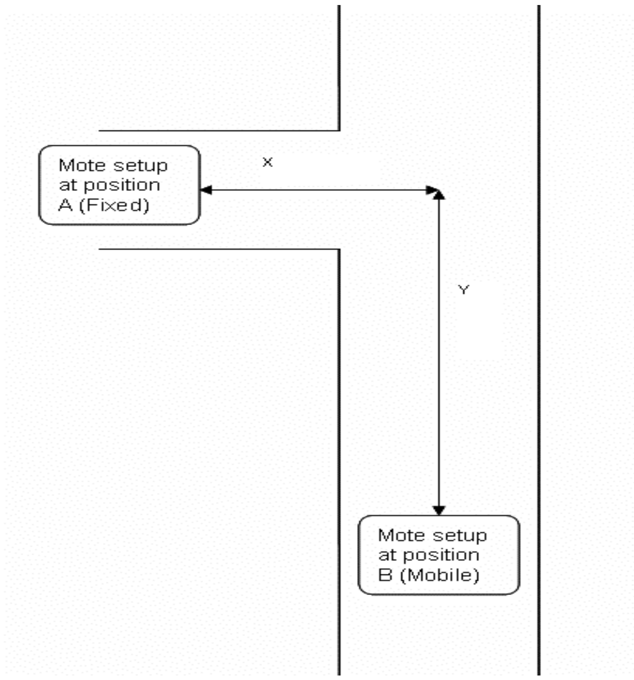


Figure 4.8 Communication with motes with no Line of Sight.

In order to simulate a handheld device which is intended to be mobile, a setup described in figure 4.3 and 4.4 is placed on a moving table and is moved in the vicinity of the circuit at position A. Thus it can be said that position A is fixed whereas position B represented a moving handheld device. In this test the motes could transmit and receive at a range of 16 meters (x+y) without LOS see figure 4.8.

Along with this a test is also made by using the "forwarder" see figure 4.9 to increase the range of transmission. A similar setup like the one described in the figure 4.8 is used for this test with one addition see figure 4.9. The larger MICA2 mote with a forwarder program is placed in the network to increase the range of transmission to 25 meters (x+y+z) by using the multi hop principle, as seen in figure 4.9.

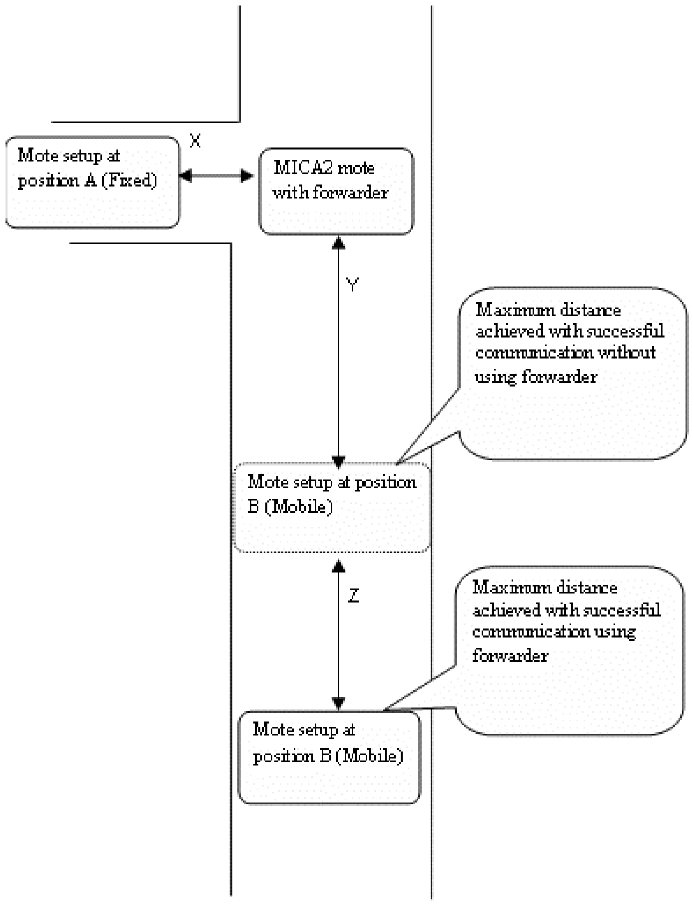


Figure 4.8 Communication among motes using a forwarder

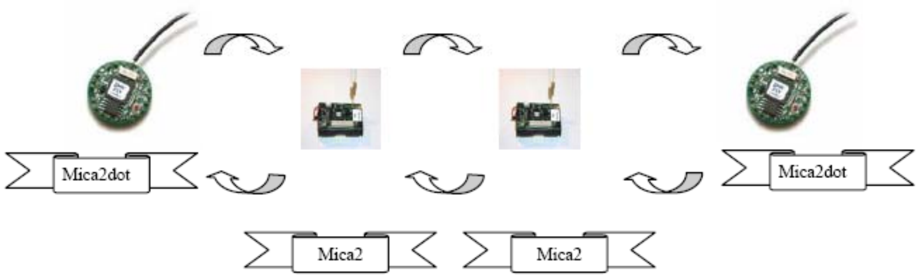


Figure 4.9 Multi hop using Forwarder installed on two MICA2 motes

Thus this test provided a possible solution to increase the range of wireless transmission. Thus by having one or more MICA2 motes with a forwarder program in a network, the range of communication could be enhanced.

5 Interviews

Interviews were made using the mockup described in the section 3.1 and 3.3 with experienced researchers from the field of rehabilitation engineering and the questions which were included in the interview have been included in Appendix B. The results of these discussions are presented here.

Figure 5.1 Mockup of the handheld device

5.1 Interview with Arne Svensk

Arne Svensk, has years of experience in rehabilitation engineering and especially in dealing with people with cognitive disabilities. The results of this discussion and his views and suggestions to the design are presented here. He suggested that instead of having keys with only color variation, each key should also have a small picture associated with it. As people with reduced cognitive abilities may forget the exact use of a key, a picture would help them remember better. As all people with such disability may not know how to read hence a visual distinction would be necessary. He also proposed messages such as “I am bored” for people who are bed ridden or people confined to small space. A message such as “Toilet break” was also necessary for a few as they may need someone to take them to the toilet. For people who are in a wheel chair a message as “I am uncomfortable” should also be included. He also suggested a real life example of people living in elderly homes, if they needed water after the nurse has just taken her hourly round; they sometimes have to wait for an hour for her to return. Hence if they had a device which would send a message as “I need water” then they can get water whenever they want. He suggested that this device would help in replacing the old clock method used in Norway. In the clock method, there are a number of small sentences where a person can place the arm of the clock, such as I need water; I am bored, Toilet break etc. He also recommended that a message sent should also be delivered to a specific person as some elderly people usually like only a selected few to help them. The concept of adding names along with the message was also appreciated by him. On this instance he gave an example of an old lady kept in hospital was being referred as room 4 bed 3 and not her name. Thus he said this device could add more personal touch to the doctor patient relationship. He said that an upgrade of this device could have a color LCD which would display an image of sender along with the message. On being asked about the physical appearance he was satisfied with the size and shape and suggested using a rubber cover on the sides to enable more grip. At this place he suggested an example of a mobile phone slipping from our hands, which is very true. In order that the device could be carried around with people he proposed using a wrap strap or a VELCRO [5.1] or a person could also place it near his seating place using VELCRO. He stressed that the device should not have sharp edges. The resemblance of the device to a remote control was also liked by him. On asked about the display, he liked the idea of the OLED LCD as it permitted reading in the dark and was also good for people with reduced vision or glasses. The green LCD was appreciated by him and he proposed it over the grey LCD. He suggested an upgrade for the design, to include a counter which would log the number of times a key was pressed. He said that this method would be useful in evaluating the service level of the staff, as there was no way to evaluate their quality of service.

5.2 Interview with Håkan Eftring

The discussion made with Håkan Eftring mainly involved elderly people. The results of this discussion and his views and suggestions to the design are presented here. In order to prevent accidental pressing of keys on holding the device he suggested that the keys should not be close to the periphery of the device. He also proposed using pictures on keys instead of key1, key2 etc. On seeing the model he suggested that the handheld device should have rounded corners and should be equipped with a strap or VELCRO so that it could be carried around. As having four keys made the device bigger he suggested that a future version could have just a few keys and a smaller LCD to make a smaller, lighter device. For future versions he suggested that the device should have a locator which would beep when you press a base switch, similar to a cordless phone. He proposed two versions of the device, a larger version with more keys may be made wall mounted and a smaller version could be made hand held. In cases of having four keys to be programmed he proposed that a suggested set of ten messages given to a user would help the user to come with ideas of four important ones to be integrated in the handheld device. A few devices could also have simple answers like yes and no programmed on keys. He also discussed a real life scenario in cases where three keys were general questions and one key corresponds to no, if a person receives a question and the answer is yes he responds by sending the same question as a positive answer else he sends “no” as a reply. A possible upgrade of having a speaker which would enable people to hear what they transmit and receive was also suggested. For people who are more used to technology he suggested using four different sounding buzzers corresponding to each key to differentiate between each message. The concept of having an antenna inside the plastic casing was also appreciated. He proposed the use of an opaque box instead of a transparent one to avoid confusion between the keys and internal circuits. He also suggested custom designs for a few people, like having the device in a shape of a cat for a person attached to his cat or having it in a shape of a car for a child and also having them in different color would help people in knowing as to which one belong to them.

6 Discussions

6.1 Suggested Modifications derived from interviews

Some suggestions which were given by Arne Svensk and Håkan Eftring as a result of interviews described in section 5.1 and 5.2 have been included here.

1) A handheld device should have pictures corresponding to each key in order to aid the user in remembering its function. Some possible suggestions which may be included here may be an image of a phone, as see figure 6.1, which may correspond to a help key on the handheld device. Any image which would enable the user to remember the function of the key could be included, like the figure 6.2 which represents two helping hands could also be included.

Figure 6.1 An image of a person using a telephone could be used to indicate a help key on a handheld device.

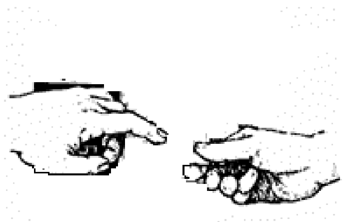


Figure 6.2 An image of two helping hands could also be included to indicate a help key.

2) The microcontroller of the handheld device could make use of a counter to determine the number of times each key is pressed.

3) There could be two versions of the handheld device, a smaller version could be carried around while a larger one could be wall mounted.

4) The handheld device could have a VELCRO along with a rubber grip, in order to aid a user in carrying it around.

7 Methods to increase transmission range between two motes

7.1 Increasing Transmit Power

The wireless transmission is achieved by making use of the RF link of the wireless sensors. These sensors can have a maximum transmission range of 60 meters with maximum power settings. But ideally for a power setting of 0x0b a range of 3 to 5 meters can be obtained. The minimum strength is 0x00 and highest is 0xff (Hex). [7.A5]

7.1.1 Enhance range by using a Forwarder

Another possible alternative is to program a few MICA2 motes with the *Forwarder* program, also developed as a part of this project, see figure 6.4. This MICA2 mote can be placed stationary in the network and will forward the RF signal to the desired location. This is another way to increase the range of wireless transmission as described in section 4.6.

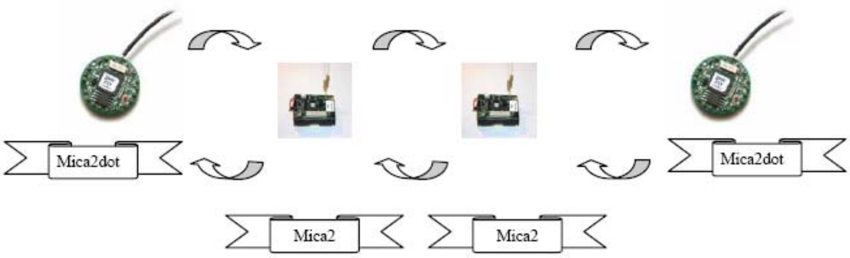


Figure 6.4 Multi hop using Forwarder installed on MICA2 motes

7.1.2 Bluetooth

The possibility of interfacing Bluetooth [7.A8] is also taken into consideration. The thought was to use the Bluetooth link of the Bluetooth transmitter instead of the RF link of the motes. The digital information could then be send over Bluetooth from one device to another. But as Bluetooth has a limited range [Personal Area Network (PAN)] [7.A9] of 1 to 10 meters for class 1 and 2 type [7.A8] transceivers, the existing RF link of the motes is considered adequate for this operation. Along with that a class 3 Bluetooth transceiver could be used which would have a range of

100 meters but as the power consumption in the class 3 device is very high, this alternative was not considered. As the aim is to develop a low power application, hence the RF link of the MICA2DOT is considered appropriate for this solution.

8 Problems encountered

A major obstacle in this thesis is the elimination of node 0 from the network, as discussed in section 1.2. Sensors are generally programmed to accept data from a specific location and forward it to node 0 to be processed. This aspect of sensors is well documented by the examples mentioned in section 2.2. Hence, in order to eliminate base station from a network a sensor had to forward the data which it received to its neighboring sensors and not the base station. So the program that had to be developed for the motes had to achieve mote to mote communication and not mote to node0 communication.

Also the program to be developed needed to achieve a two way communication unlike the usual application of sensors which achieve a simplex communication as seen in section 2.2, hence the program that had to be developed for the motes had to have two separate functions while transmitting and receiving. Thus the program for the motes had to achieve a half duplex communication which would then be used by the handheld device to send and receive messages.

Initially it was thought of using the I/O pins on the MICA2DOT mote to send and receive information but the limited number of I/O pins on the motes made this a restricted solution in terms of the number of devices which could be interfaced. Also when the I/O pins were configured as input the microcontroller of the MICA2DOT mote assigns them as high which prevented them from accepting any input, thus a restricted solution could have been obtained using I/O pins of the MICA2DOT motes. Hence an alternative solution had to be used which would be reliable and flexible in the number of keys being used. Hence, the possibility of using the ADC pin as an input is considered. It was thought that each key on the handheld device would correspond to a separate voltage which could be given to the ADC pin, which in turn would convert it to a distinct 16 bit value, thus each key on the handheld device corresponds to a different 16 bit value. This 16 bit value would be received on the receiving mote and thus by using this value a combination of three or more I/O pins which had been configured as output could be made high or low. Thus every key on the handheld device at position A corresponded to a different combination of I/O's at position B. By making use of the ADC pins seven keys could be used, which is also necessary in some cases of upgrades.

In order to achieve a half duplex communication a program TRIA is developed and installed on two motes in a network. Another obstacle is the possibility of increasing the range of transmission. In this aspect many possible solutions were considered and description of each of them has been included in the section 4.4.4 but the use of the forwarder is considered to be suitable.

9 Conclusion and future work

9.1 Conclusion

It has been possible to solve the problem of eliminating node0 by making use of the program developed as the part of this thesis work. The tests which were conducted in the section 4.4 further support my claim of achieving a two way communication without using base station. The limitation of having two messages has been overcome and the software developed for the motes could send and receive up to seven different messages. The range of transmission among motes can be extended by making use of a MICA2 mote with a forwarder installed on it. Along with this it can also be concluded that there could be more than two motes in a network. Thus, more than two individuals could send and receive messages. Also it can be concluded that the mockup that was used to generate feedback for the development process, further strengthens the possibility of having a handheld device which would enable a user to send and receive messages.

9.2 Future work

One aspect which could be explored in the future could be to interface motes to a mobile phone which would enable motes to communicate with each other over hundreds of meters. Along with that, the option of adding sound to a handheld device is one field which can be looked into in the future. Along with text messages, a small speaker could be interfaced which would enable a receiver to hear the message which has been displayed on his user interface. Thus this would prove an added bonus for users with reduced vision.

10 References

[1.1] Motes
1) Motes, Smart Dust Sensors, Wireless Sensor Networks
www.xbow.com/Products/Wireless_Sensor_Networks.htm

2) Wireless Integrated Sensor Network (WINS)
http://portal.acm.org/citation.cfm?id=332838&coll=portal&dl=ACM&CFID=66298191&CFTOKEN=71385218

[1.2] Markus Bylesjö
http://www.certec.lth.se/dok/trygghetskommunikation/

[1.3] Certec
Center for Rehabilitation Engineering Research, Lund Institute of Technology
http://www.certec.lth.se/

[1.4] MICA2 – MICA2 Mote
http://www.xbow.com/Products/productsdetails.aspx?sid=72
[1.4A]MICA2 Mote Datasheet
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf

[1.5] TinyDB
A declarative database for sensors
http://telegraph.cs.berkeley.edu/tinydb/

[1.6] Windows Net meeting
Windows® software for file sharing and audio conversation
http://www.microsoft.com/windows/NetMeeting/default.ASP

[1.7] Half duplex
Two way data transmission standard in radio communication
http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci212221,00.html

[1.8] LEDS
Light Emitting Diodes
http://www.theledlight.com/

[1.9] LCD
Liquid Crystal Display
http://www.eio.com/lcdintro.htm

[1.10] GSM
Global system for mobile communication
http://www.gsmworld.com/index.shtml

[1.11] multi hop
http://www.ccs.neu.edu/home/rraj/Talks/MultihopTutorial/

[1.12] Universal Design
http://design.ncsu.edu/cud/

[1.13] ad hoc - network
http://searchmobilecomputing.techtarget.com/sDefinition/0,,sid40_gci213462,00.html
http://en.wikipedia.org/wiki/Ad_hoc

[1.14] Simplex
One-Way data transmission standard in radio communication
http://cbdd.wsu.edu/kewlcontent/cdoutput/TR502/page45.htm

[1.15] MICA2DOT- MICA2DOT Mote
http://www.xbow.com/Products/productsdetails.aspx?sid=73

[1.15A]MICA2DOT Mote Datasheet
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf

[1.16] mockup
The New Media Reader by Janet H. Murray and Lev Manovich. ISBN 0-262-23227-8

[1.17] Håkan Eftring - Administrative head of Certec
http://www.english.certec.lth.se/staff/hakane/

[1.18] Arne Svensk - Instructor, doctoral student at Certec
http://www.english.certec.lth.se/staff/arne/

[2.1] Crossbow
Crossbow Technology - Wireless Sensor Networks, Inertial & Gyro systems and Smart Dust
http://www.xbow.com/
[2.2] network topology
http://www.homenethelp.com/network/

[2.3] cell phone
http://cellphones.about.com/

[2.4] Personal Digital Assistants (PDA)
http://www.pdastreet.com/

[2.5] 802.11 Wireless Local Area Network
http://www.ieee802.org/11/

[2.A1]-
http://www.sensicast.com/

[2.A2]- Sensors in news
http://news.bbc.co.uk/1/hi/technology/4618086.stm

[2.A3]- Sensors in news
http://news.bbc.co.uk/2/hi/science/nature/4736196.stm

[2.A4]- Sensors in news
http://news.bbc.co.uk/1/hi/world/asia-pacific/4749489.stm

[2.6] Embedded Wireless Sensor Networks
http://webs.cs.berkeley.edu/

[2.7] TinyOS (TOS)
TinyOS is an open-source operating system designed for wireless embedded sensor networks.
http://www.tinyos.net/

[2.8] Atmel
Atmel Corporation
http://www.atmel.com/

[2.9] ATmega128L
http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf

[2.10] MPR500CA
Sequoia Technology Ltd
http://cnb-host4.clickandbuild.com/cnb/shop/sequoia?listPos=&op=catalogue-products-null&prodCategoryID=24

[2.11]UART
Universal Asynchronous Receiver/Transmitter
http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-uart/

[2.12] nesC
A Programming Language for Deeply Networked Systems
http://nescc.sourceforge.net/

[2.A7]- TinyOS
http://www.tinyos.net/special/mission

[2.13] TinyOS concurrency model
System Architecture Directions for Networked Sensors
http://www.tinyos.net/papers/tos.pdf

[2.14] Cygnus Solutions
http://cygnus-solutions.com/

[2.15] POSIX systems
Portable Operating System Interface, with the X signifying the Unix heritage of the API
http://standards.ieee.org/regauth/posix/

[2.16] Linux
Open-source computer operating system
http://www.linux.org/

[2.17] API
Application programming interface
http://en.wikipedia.org/wiki/API

[3.1] OLED
Organic Light Emitting Diode
http://www.universaldisplay.com/tech.htm

[3.1A] Voltage divider
http://www.aikenamps.com/VoltageDividerRule.htm

[3.2] SMS
Short Messaging Service
http://smsforum.net/

[4.1] RF
Radio Frequency
http://www.sengpielaudio.com/calculator-wavelength.htm

[4.2] Line of sight (LOS)
http://cnx.org/content/m0538/latest

[5.1] VELCRO

http://en.wikipedia.org/wiki/Velcro

[7.A5] MICA2DOT Datasheet
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf

[7.A8]- Bluetooth
www.bluetooth.org

[7.A9]- Personal Area Network
http://en.wikipedia.org/wiki/Personal_area_network

11 Appendix

11.1 Appendix A

11.1.1 Quick Guide to mote Programming

This guide is meant as a quick reference for parameter settings and mote programming for users that have no experience in TinyOS. It also includes a quick demonstration of some applications that comes with the TinyOS installation file. Note that more detailed instructions are available in the TinyOS directory after installation.

11.1.1.1 Set-up

After installing TinyOS, which takes up to an hour, use Cygwin to change directory (cd) to \cygwin\opt\tinyos-1.x\tools\scripts directory and type "toscheck". To do this, type the following:
cd root, cd opt, cd tinyos-1.x, cd tools, cd scripts, toscheck
TinyOS will perform a system check and return a message "toscheck completed without errors".

11.1.1.2 Radio Configuration

Initially for basic programs which do not make use of the RF link of the mica motes (either MICA2 or MICA2DOT), the Radio frequency parameter dose not play a significant role. The radio frequency has to be set depending on the type of motes that are being used; these motes can be set for frequencies between 315 and 915 MHz.

11.1.1.3 Setting the radio frequency

Here is the procedure to modify the frequencies in CC1000Const.h. The user should check the motes to determine which frequency they operate with. E.g. a 900 MHz mote will not work when it is set to 433 MHz but it works fine when set to 914.077 or 915.998 MHz. The CC1000Const.h is located in \cygwin\opt\tinyos-1.x\tos\platform\MICA2
Open CC1000Const.h and change line 212 to the required frequency. Here it's set to 914.077 MHz.

```
211  #ifndef CC1K_DEF_PRESET
212  #define CC1K_DEF_PRESET          (CC1K_914_077_MHZ)
```

```
213  #endif
```

A list of available frequencies in MHz is given below:

315_778
433_002
434_845
914_077
915_998

11.1.1.4 Setting the radio transmission strength

Here I have included a way in which the variable for transmission strength in MICA2 and MICA2DOT motes can be set. By changing the transmission strength, a range of approximately 0 to 60 meters can be obtained. The minimum transmission strength is 0x00 and maximum is 0xff (Hex). [1.15A]

Open the CC1000Const.h file and find the table with the frequency you previously changed to. Here's an example for 914_077_MHZ

```
// (3) 914.077 MHz channel, 19.2 Kbps data, Manchester Encoding, High Side LO
 { // MAIN   0x00
   0x31,
   // FREQ2A,FREQ1A,FREQ0A  0x01-0x03
   0x5c,0xe0,0x00,
   // FREQ2B,FREQ1B,FREQ0B  0x04-0x06
   0x5c,0xdb,0x42,
   // FSEP1, FSEP0     0x07-0x8
   0x01,0xAA,
   // CURRENT (RX MODE VALUE)   0x09 (also see below)
   ((8<<CC1K_VCO_CURRENT) | (3<<CC1K_LO_DRIVE)),
   //0x8C,
   // FRONT_END  0x0a
   ((1<<CC1K_BUF_CURRENT) | (2<<CC1K_LNA_CURRENT)
| (1<<CC1K_IF_RSSI)),
437     //0x32,
438     // PA_POW  0x0b
439                        ((0x8<<CC1K_PA_HIGHPOWER)      |
(0x0<<CC1K_PA_LOWPOWER)),
440     //0x7f,
```

Change line 439 and 440 to:

```
439                      //((0x8<<CC1K_PA_HIGHPOWER)       |
(0x0<<CC1K_PA_LOWPOWER)),
```

```
0xff,
```

0x7f is the hex value that determines the transmission strength. The lowest and the highest values are 0x00 and 0xff respectively.

11.1.1.5 Setting the Group id and node id

11.1.1.5.1Group id

Open the Makerules file found in the apps catalog (\cygwin\opt\tinyos-1.x\apps). Change to preferred local group id in line 30.

```
29  ifndef DEFAULT_LOCAL_GROUP
30  DEFAULT_LOCAL_GROUP := 0x7d
31  endif
```

The group id is used for communication between motes. Two motes with different group id will not communicate with each other, even if they use the same frequency. The default group id is set to 0x7d. Any value between 0x00 and 0xff (Hex) can be used for setting a group id.

11.1.1.5.2Node id

Node id corresponds to the motes id number in the network. Motes with the same group id are identified within the group by the node id; therefore each node that uses the same group-id must have a unique node id. Nodes with different group id can have the same node id. Node id numbers 255 and 126 are reserved and should be avoided.

Some Linux commands

cd apps	change to apps directory
cd ..	Jump up a directory (place space between cd and ..)
cd /	jump up to the root directory
dir	displays the directories and files in the current directory
ls	displays all files and directories in the current directory

The tab key is useful for using short terms of the directory names, e.g. when in the "opt" directory, type "cd tin" followed by the tab key. The line should say cd tinyos-1.x/.

11.1.1.6 MIB510 Serial Interface Board

The MIB510 interface board is a multi-purpose interface board used with the MICAz, MICA2, MICA, and MICA2DOT family of products. It supplies power to the devices through an external power adapter option, and provides an interface for a RS-232 Mote serial port and reprogramming port.

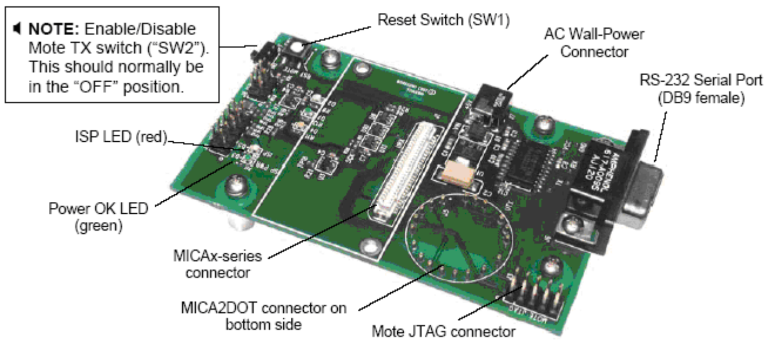


Figure A.1 MIB510 programming board

The MIB510 has an on-board in-system processor (ISP)—an Atmega16L located at U14—to program the Motes. Code is downloaded to the ISP through the RS-232 serial port. Next the ISP programs the code into the mote. The ISP and Mote share the same serial port. The ISP runs at a fixed baud rate of 115.2 kbaud. The ISP continually monitors incoming serial packets for a special multi-byte pattern. Once this pattern is detected it disables the Mote's serial RX and TX and then takes control of the serial port.
WARNING: Some USB to DB9 serial port adapters cannot run at 115 kbaud.
The ISP processor is connected to two LEDs, a green LED labeled "SP PWR" (at D3) and a red LED labeled "ISP" (at D5). SP PWR is used to indicate the power state of the MIB510. If the ISP LED is on, the MIB510 has control of the serial port. It will also blink once when the RESET (SW1) button is pushed and released.

11.1.1.7 Programming Concepts

Warning!!!! If the programming board has an external power supply always switch off the motes battery before connecting them in the programming board.

Warning!!!! The oscillator part on the radio is very sensitive and might interfere and damage each other if the motes are placed within 2 feet from each other.

11.1.1.8 Installing an application on a mote

Open a Cygwin shell, cd to opt/tinyos-1.x/apps. cd to a desired program e.g. CntToLeds that counts up an integer and displays the three lowest bits on the leds. A simple application for testing the radio transmission is to program one mote with CntToRfm and one with RfmToLeds. Turn both on and you should be transmitting the integer value from the counter to the receiver.
For installing a program on to a mote go to the directory where the program has been saved. E.g opt/tinyos-1.x/apps/CntToLeds

For MICA2DOT type in:
"MIB510=/dev/ttyS0 make install.<node id> MICA2DOT"

For MICA2 type in:
"MIB510=/dev/ttyS0 make install.<node id> MICA2"

The ttyS0 in the line above interfaces the board to the computer via the COM port 1. If you're using COM port 2 type ttyS1 instead of ttyS0.

Example:
To program a MICA2DOT mote with node id 1 and programming board is connected to com1, type:
"MIB510=/dev/ttyS0makeinstall.1MICA2DOT"

11.2 Appendix B

11.2.1 Questionnaire

What do you think about this device?

How effective would such a device be?

Which kind of people would use such a device?

Will people appreciate this technology or will be intimidated by it?

What do you think of the wireless sensors and their use in this project with respect to size and power consumption?

What do you think of the OLED LCD?

In which scenarios could this device be used?

Would adding the name of a person along with the message would give this product a personal touch?

What do you think of the physical appearance of the prototype?

Are the keys on the user interface placed fairly apart from each other?

What suggestions can you provide for further upgrades of this system?

11.3 Appendix C

Hardware.h: Before I describe the program I would like to describe the hardware.h file of the MICA2DOT motes. This file is used to initialize the ports for the microcontroller of the MICA2DOT and is also used to set pins as input or output. In the hardware.h file of MICA2DOT an ADC6 pin was set as input and PW0, PW1 and INT0 were set as output ports, as seen in figure 2.4.

The flowchart of the TRIA program has been represented here. As seen in figure C.1, the program is a combination of four subprograms which in turn make use of other functions.

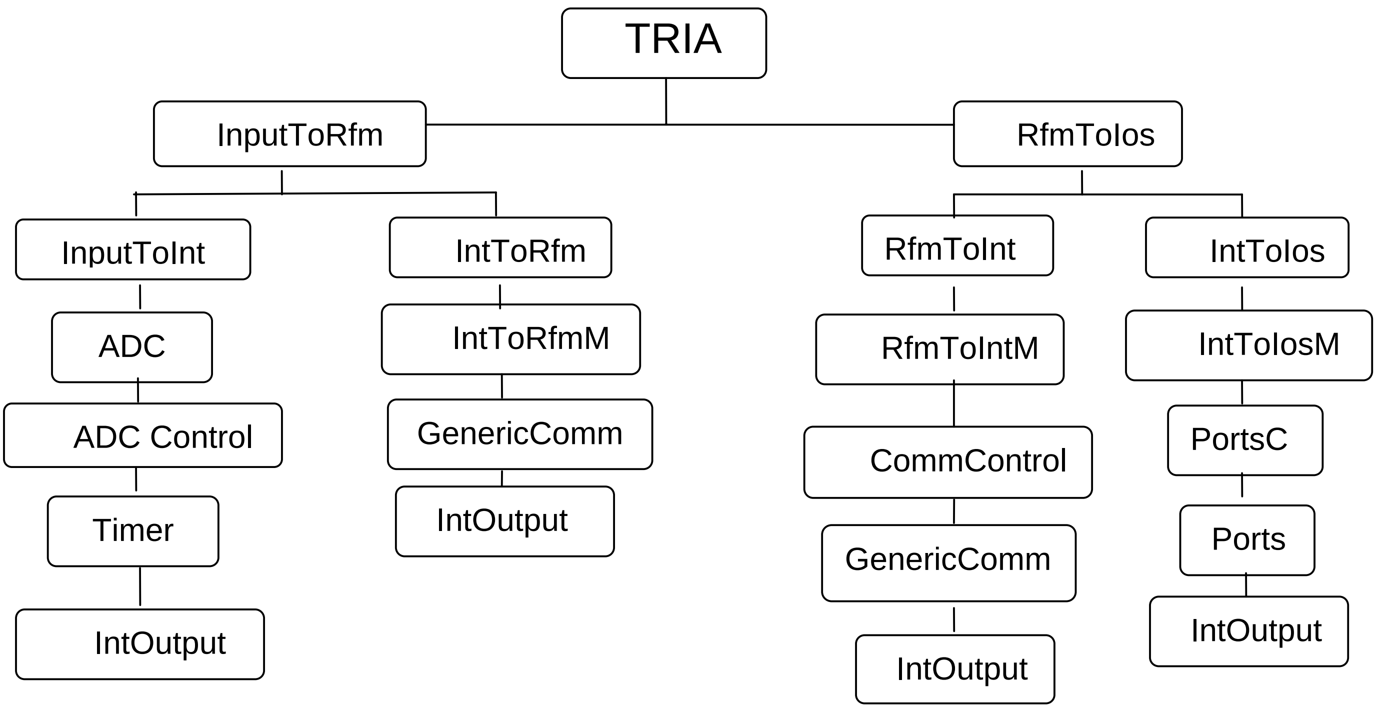


Figure C.1 Flowchart of TRIA

A brief description of the above program is given here.

InputToRfm: This is the first module we come across in the program and is also the most important one. InputToRfm was designed to accept an input voltage and using the ADC pin, which was checked every second. InputToRfm made use of two modules InputToInt and IntToRfm. InputToInt converted the received input voltage into intergers and IntToRfm transmitted the interger values over the RF link.

ADC: InputToInt in turn made use of the **ADC** module which initiates ADC conversion at an assigned pin and converts the input value to a 16bit value.

ADCControl: ADCControl is another module which determines the sampling rate at which the data should be sampled.

Timer: InputToInt also make use of a **Timer** module which could be a 'TIMER_REPEAT' or a 'TIMER_ONE_SHOT' timer. A timer of type TIMER_REPEAT can be fired a number of times but a TIMER_ONE_SHOT can be fired just once.

```
command result_t start(char type, uint32_t interval);
```

In this statement the char type would indicate the type of the Timer such as TIMER_ONE_SHOT or TIMER_REPEAT and the interval indicates the duration of the timer. This program made use of the TIMER_REPEAT timer.

IntOutput: The **IntOutput** module determines as to which module provides the input and which one would get the corresponding output.

InputToInt.IntOutput -> IntToRfm; This statement will make sure that the result obtained from the InputToInt module is given to the IntToRfm module, in order to transmit it using the RF link.

IntToRfm: The IntToRfm module makes use of **Genericcomm** in order to initialize the Radio Control unit and the UART Control unit. The **Genericcomm** module accepts the 16 bit value from the **InputToRfm** and transmits it using the RF link.

IntToRfmM: Along with Genericcomm, IntToRfm also made use of the **IntToRfmM** module. IntToRfmM made use of a boolean variable "pending" to check for the status of the data being transmitted. If pending was true then the information is transmitted at the next instant using the LOCAL_ADRESS of the motes else if pending is false no data is left to be transmitted.

Timer: Timer is used to set the clock frequency.

RfmToInt: RfmToInt makes use of the **RfmToIntM** and **Genericcomm** .

RfmToIntM: **RfmToIntM** makes use of **GenericComm** and **CommControl** to receive data.

CommControl: CommControl is used to check the value of the CRC check flag. If set to 'true', all received packets are passed to the receiver. If set to 'false', only packets destined for this node (or sent to the broadcast address) are passed on (multi hop)

IntToIos: This module is used to set or reset a few output ports on the destination (MICA2DOT). In order to achieve this **IntToIos** uses the **InttoIosM, Ports** and **PortsC** files.

PortsC: This module is used to initailise, set or reset a required port. It uses functions such as Ports.init(), Ports.Port1set() and Ports.Port1off() to initialize, set and reset Port1.
Ports: Ports makes a number of function calls to the functions declared in the PortsC.
IntToIosM: **InttoIosM** use the **Ports** and **PortsC** file to initialize, set or reset ports. It performs a bitwise AND operation on the received 16 bit value so as determine the sequence of ports which can be set or reset.

A brief explanation of the Forwarder program has been presented here.

RfmToRfm: RfmToRfm program also called as Forwarder was created with the intention of having a multi hop network. When this program is installed on a MICA2 mote and placed in a network it would forward the RF signal it received to the neighboring motes thus to increase the range of transmission.

Certec reports

A selection of reports:

Doweyko Joanna; Sandberg Bibi,
EYESEE - An RFID equipped identification tool for people with visual impairments, 2005

Hallengren Erik; Hed Ingrid,
A Simplified User Interface for Mobile Phones, 2004

Lindmark Louise; Söderberg Kristina,
The Certec Clock in the palm of your hand: a portable, digital time aid for people with cognitive limitations, 2004

Rowell Sara,
Peripheral motion detection in traffic simulation, 2004

There is a need for quick exchange of short messages. These messages could be day to day messages like "How are you?", "Do you want to go out", "I am bored" etc. Many times it has been seen that a standard set of messages is often repeated in a conversation. The goal with this thesis was to suggest a device which could be used to send and receive standard messages. In order to make the device mobile wireless sensors were used to form a wireless link. The aim was to develop an ad hoc network which would be independent of the base station.

This document is also available on the Internet:
www.certec.lth.se/doc/awireless

Department of Electroscience
Faculty of Engineering
Lund University

Certec
Division of Rehabilitation
Engineering Research,
Dept. of Design Sciences
Faculty of Engineering
Lund University

Certec, LTH
Box 118
221 00 Lund
Sweden

Sölvegatan 26
Lund, Sweden

+46 46 2224695

+46 46 2224431

lena.leveen@certec.lth.se

http://www.certec.lth.se

Department of Electroscience at the Faculty of Engineering at Lund University specializes in Circuit Design, Radio Communication, and Signal Processing. The department, through its 60 employees, 30 Ph.D students and more than 60 masters' students strives to enhance existing technologies and lay the foundation for future research in the field of Circuit design and Radio Communication. The annual budget is approximately 50 million Swedish kronor.

Certec is a division of the Department of Design Sciences at Lund University.

The aim of our research and courses is to contribute to improved opportunities for people with disabilities by developing more usable technical devices, new design concepts, and new individual methods of learning and searching.

Certec employs about 25 people. Our annual budget is approximately 15 million Swedish kronor.

MASTER'S THESIS CERTEC, LTH NUMBER 1:2006
MARCH 2006

Vipul Surve

A Wireless Communication Device for Short Messages