YORK UNIVERSITY

FACULTY OF SCIENCE AND ENGINEERING

ENG 4000

ENGINEERING PROJECT

INTERIM REPORT

TABLE OF CONTENTS

WiDAR

Wireless Detection and Ranging

Abstract

Wireless Sensor Networks (WSN) is emerging as an exciting and innovative technology, one which knows no bounds. There is currently a great deal of research in the WSN field, this can be attributed to its large scope involving issues which are both theoretical and practical in nature. This led our group to purse this subject for our engineering thesis project.

This report outlines the progress of the project and its current standing in comparison to the initial objectives and desired outcomes. Setbacks that have been faced and their effects on the project are described in detail as well. Wireless Sensor Networks rely on their ability to conserve power to ensure their longevity and autonomy. Consequently, their ability to appropriately react to adverse circumstances is paramount. One such example would be that of a node failure somewhere along the routing path. Typically, this may sever all communications between the end source and the base station. To alleviate this dilemma we propose the incorporation of a mobile robot integrated with a node. The robot is designed to find the best location with the network of nodes to enable communication to resume.

Team Members and Area of Focus

Members:

Hamdi Roumani
Team Leader
taal@yorku.ca
Focus: *Responsible for comparing and evaluating candidate wireless sensor network kits based on a criterion stated in the "Short Project Description". Once this decision is taken this member will work along side Douglas Stamp on the communication between the wireless sensor nodes. The project leader will act as a liaison between the WIDAR team and the advisor.*

Patrick Tayao
patrick.tayao@percepta-crm.com
Focus: *Researching alternative methods of interfacing a mobile robot with wireless sensor nodes.*

Douglas Stamp
dstamp@yorku.ca
Focus: *Website design and maintenance. Researching communication methods between wireless sensor nodes.*

Tyson J Hamilton
tyson@yorku.ca
Focus: *Website design and maintenance. Researching communication between wireless sensor nodes and the mobile robot (Working along side Patrick Tayao).*

Faculty Advisor

Professor Natalija Vlajic
Department of Computer Science and Engineering
vlajic@cse.yorku.ca
416-736-2100 x 77878

Introduction

Wireless Sensor Networks is a relatively new as well as exciting computer technology that offers a plethora of possible applications to both everyday and future computing solutions. From senor based monitoring and tracking in remote locations such as rain forests, to communication between autonomous robotic devices designed to manipulate an environment, wireless senor networks is an area of research rapidly developing and garnering much interest.

These networks essentially consist of two components: nodes and a base station. Nodes are low power consuming devices that are integrated with data feeds to monitor a situation. These feeds may contain environment sensors mounted to different nodes, which are called source nodes. The information gathered by the nodes is then transmitted from one to another using peer-to-peer protocols. The base station provides a powerful communication link within the network of nodes. The node integrated into the base station is called the sink node, through which all collected information propagated from different nodes is relayed to an end user.

If a node or nodes within the network lose communication with a node sending or receiving data, a break within the sensor network is created. The communication network will then consist of two unlinked sub networks. Therefore the base station will not be able to retrieve and transfer data collected by a sink node to the end user.

The proposed solution to such a problem is integrating a mobilized senor node, via a mobile robot, into the severed network. The mobile unit will act as a communication bridge between the two independent sub networks of nodes. This solution will be implemented using a hole detection algorithm. The algorithm is essentially the method in which the mobile robot node will search for and then eliminate the communicational gap or 'hole' between the sub networks, thus reestablishing the original unified wireless sensor network. A challenging task in the implementation of this algorithm is the localization of nodes to ensure proper placement of the mobile robot. Since nodes are scattered in different geographical locations the localization process is difficult.

To facilitate the management of the several aspects of this project, the solution implementation has been split into two parts: One for dealing with all aspects of the mobile robot and how it will be integrated into the WSN system; and the second part applying to the programming of nodes, using the aforementioned algorithm, to assist the robot in locating and eliminating the communication gap. Our team of four members divided itself into subgroups of two, with each subgroup have primary responsibility over the implementing one of the two parts of the project.

Mobile Robot

Technical Description

Overview

The mobile robot will be responsible for bridging the gap between two disconnected groups of motes. In order to accomplish this, there are two major components that will be addressed: robot control and network communication. To address these requirements, the mobile robot will be using an HC12 Dragon12 board. This device was selected to control the robot because of its numerous features and processor speed, including a Pulse Width Modulator (PWM) and an IIC component. The PWM component is required to control mobility of the robot on the fly. By specifying the duty cycle, the motor speed can be increased or decreased as each motor goes in and out of sync. The PWM will be connected to an H-Bridge since the HC12 does not supply enough current in its output ports to drive each motor. The HC12 will produce its outputs to an H-Bridge and the H-Bridge will be powered up and produce the required current to drive the motors. Communication to the WSN will be established by the use of an IIC connection to one of the provided motes. This mote will act as a relay for the commands provided by the base station and responses sent by the mobile robot to the base station. Since IIC in both devices do not have pull up resistors, this will need to be designed to allow the receivers to recognize the signals.

Mobility

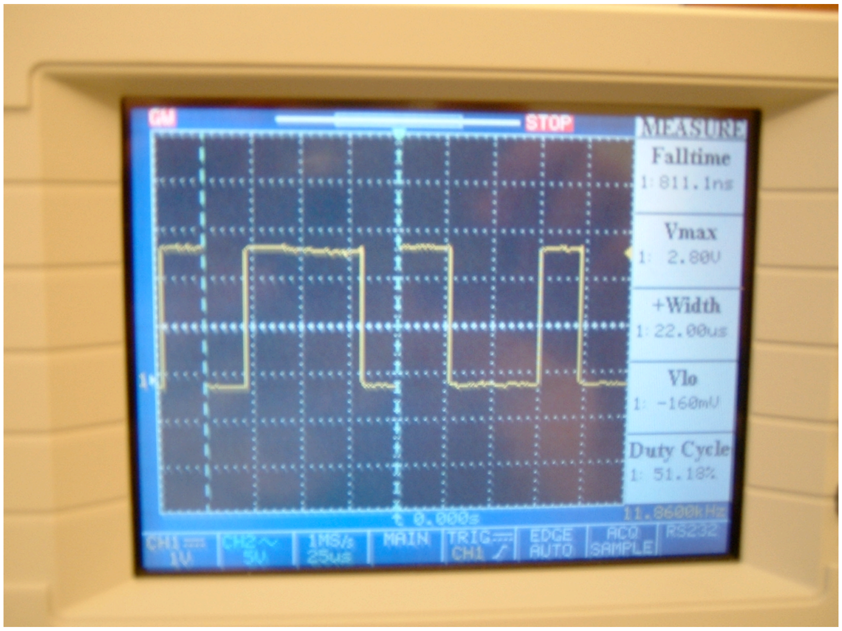
The mobility of the robot is controlled by the HC12's pulse width modulator (PWM) system connected to an H-Bridge. The system uses a Texas Instrument SN74410NE H-Bridge to allow enough power to drive each motor with a 7.2 V output. The system's PWM provides the ability to change motor speed on the fly depending on the rate of rotation in of each wheel. Each wheel contains a spoke encoder wheel and a shaft encoder to sense the speed that each wheel rotates. Depending on the part of the wheel that is facing the infrared sensor of the shaft encoder, it will output either logic one or zero. To power up the shaft encoder, a voltage divider is used to split the voltage from our 7.2 V battery to output 5 Volts for the encoder input. Each transition from 1 to 0, or vice versa, represents 1/30th of a rotation of the wheel. The system will monitor the speed of rotation and adjust each PWM system according to synchronize its rotation. Currently, the connections from the HC12 to the motor and the H-Bridge has been designed and implemented on code. The next implementation is the system's shaft encoder. Initially, this was not going to be implemented. Due to issues in guiding the robot and discussing our issues to professors, it was deemed that this was required. This implementation is currently on the way. Parts have been ordered and initial design has been completed. The weekend of the 19th of January should complete the implementation and be available for testing and calibration.

Robot to Mote Communication

To establish communication from the robot to the Wireless Sensor Network, the IIC port in the HC12 will be connected to a WSN node. The WSN node will then act as a relay to provide information from the network to the mobile robot. The clock and data in the HC12 will be generated in Port J6 and J7 and a pull up resistor will be used to settle the output signal at up to 5 V. The registers required to drive the signal has been determined and it will operate in wait mode until a signal from the mote is received. At this time communication between the mote and robot will commence. All work on this connection is currently theoretical. One of the issues that might arise is synchronizing the clock speed to ensure that the HC12 will be able to read data at the same transmission rate of the Mica2 and vice versa. Since the Mica2 is much slower than the HC12, the HC12's clock will need to be divided to about 25 KHz.

Mica2 I2C Connection

The IIC connection in the mote will be used to communicate with the robot. The initial testing of the I2C connection with the mote has been completed. The initial testing consisted of learning the TinyOS code and creating a program to send an output to its IIC port. The code has been created (see page xxx) and output was analyzed using an oscilloscope. The output produced an output of 1.4 V for a logic of one and an output in the order of millivolts for logic 0. The clock was determined within the range of 24 Khz. Because this output is low, a pull up resistor is required to produce the proper voltage level.



To complete this communication between the Mica2 and the HC12, All that is needed is connecting the outputs to a pull up resistor and testing synchronization issues in data send and receive.

Output of Oscilloscope

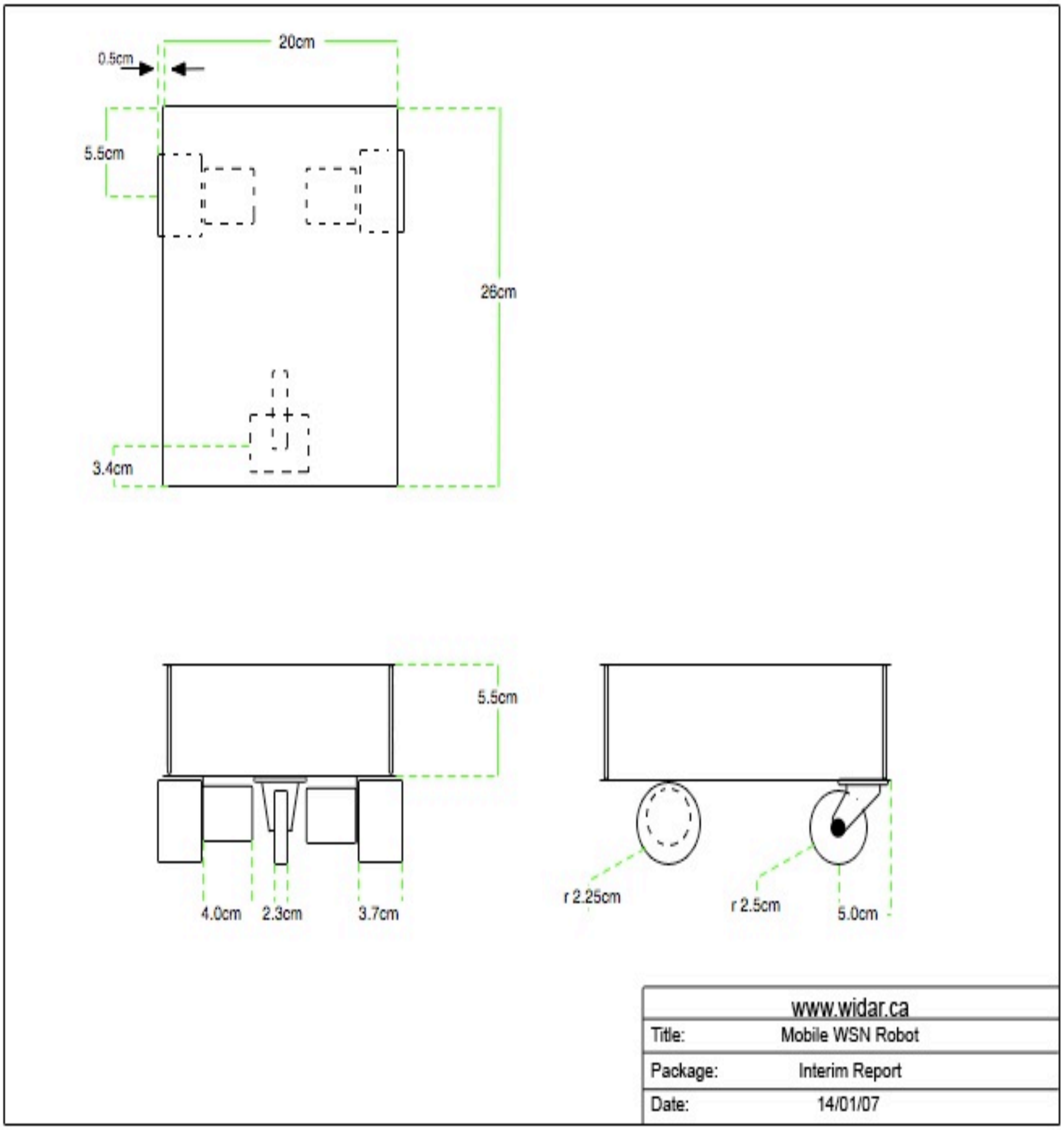
Mobile Robot Construction

Analyzing our desired performance goals and minding all physical and power constraints guided the design of the robot. The robot is able to process simple commands, having the ability to move forwards, backwards and turn by variable amounts. The physical constraints involve the driving power of the motors that are used, which limits the weight of the robot. The power constraints are defined by the limitations of the battery used and effects the electronic devices involved.

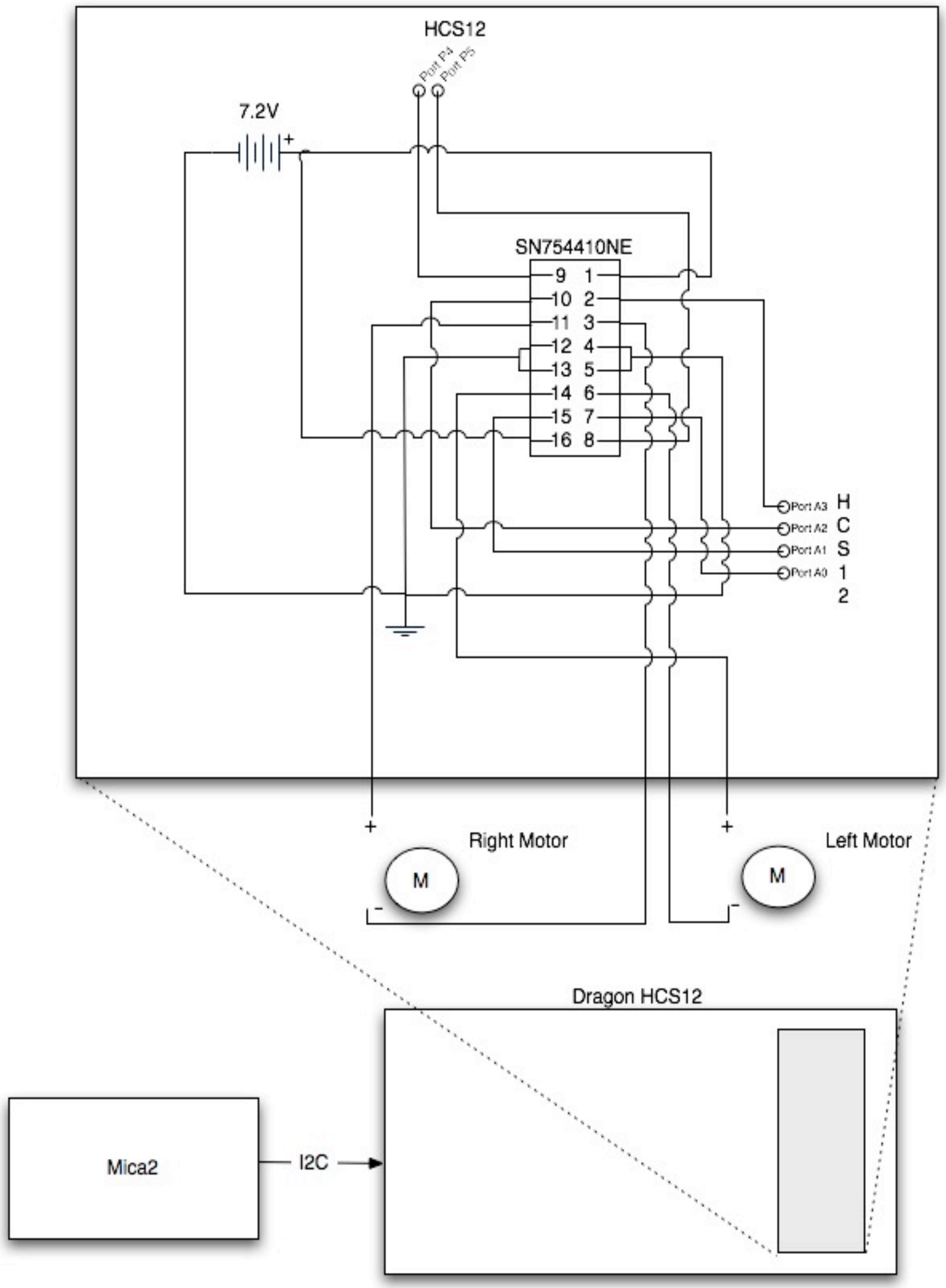
The robot consists of a two layer chassis that is illustrated in the orthographic projection within this document. The bottom layer is made using a lightweight metallic alloy sheet, although heavier than the top layer provides greater durability and strength. The bottom layer has the motor mounts secured beneath it, which support the motors that are driving the wheels. These urethane wheels have been tested to push a payload of over 14lbs and are situated at the "front". At the "rear" of the robot, beneath the bottom layer is a castor wheel. This castor wheel simplifies the design of the robot for turning because synchronization is only required between the two driving wheels, it also allows us to save weight by using only two motors for the entire system. The bottom layer also provides housing for the battery. Bolts that are secured to the bottom layer support the upper layer. This layer is constructed using sheets of plastic, which provide the desired strength and are very lightweight. The upper layer provides a means of securing the Dragon HCS12, and the MICA2 that are being used. Placing the MICA2 on the highest layer will allow better communication strength, and less interference.

The following pages contain schematic diagrams of the mobile robot.

Mobile Robot Schematics



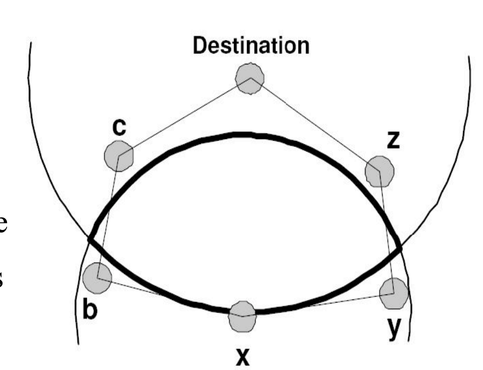
Mobile Robot Circuit Diagram



Proposed Algorithm for Hole Detection

The central focus of this project as addressed in the introduction is too, using a mobile robot, enable communication between the source and sink when a mote on the packet routing path fails. A key component to this problem is the algorithm used to place the mobile robot in the correct position. Working alongside Prof. Vlajic, Nelson Moniz and Michael Portnoy a few possible algorithms were proposed. The packet forwarding routine used by the WSN relies on a greedy algorithm (shortest distance) this can lead to the following problems:

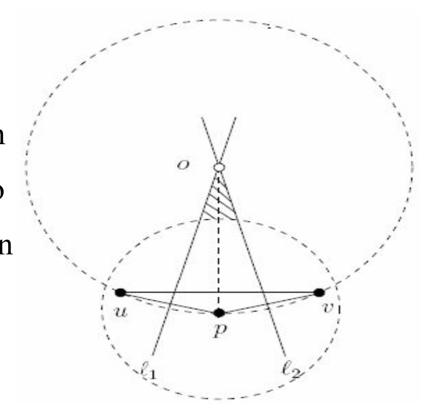
Figure 1



Since nodes forward packets to nodes which are closer to the destination of the sink node than themselves, if a node which received data is the closest node to within its range of communication in the network to the sink node, it will determine *itself* to be the next destination node to forward the data to. This creates an infinite loop, and consequently a hole in the senor network's node communication. Figure 1 illustrates this, with 'x' being the closest node, in its range of communication, to the Destination node, so it does not forward to either the 'b' or 'y' nodes.

To detect the hole the following algorithm can be used:

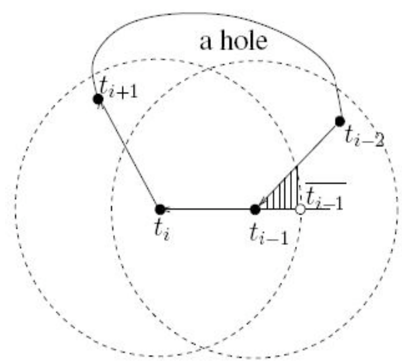
Figure 2



Certain nodes in the network form the boundary of the communication hole. These nodes are referred to as Stuck Nodes. In Figure 2, 'o' is the Destination or Sink Node. 'p' node is determined to be a Stuck Node if the angle subtended by 'u', 'p' and 'v' is greater than 120 degrees, with the perpendicular bisectors of 'u->p' and 'p->v' both passing through 'o'. If the subtended angle is not greater than 120 degrees, this implies that nodes 'u' and/or 'v' must be closer to 'o' than node 'p'. If not, 'p' is closest to 'o'.

Finally, to map the boundary of the routing hole:

Figure 3



We use the BOUNDHOLE algorithm, which starts at a Stuck node say t_i. The algorithm traces through locating the closet other stuck node to t_i, which we'll call t_{i+1}. This continues recursively until the algorithm relocates t_i, in a closed cycle of Stuck nodes, in other words, $t_{i+n} = t_i$. The boundary formed by tracing the path formed by the Stuck nodes is the Network communication hole as shown in figure 3.

Using the two algorithms discussed above the following three algorithms can be used:

Mapped boundary method

All nodes in the network are localized having an x, y, position values. When the sink node loses feed information, it floods the network with a signaling message requesting all nodes to run the TENT algorithm, determining which nodes are Stuck nodes. If a node determines itself a Stuck node, it sends out a beacon to the Sink node (also referred to as the Base Station). The sink passes this information to the node integrated into the Mobile robot, which is then deployed from the Sink. The robot travels in a straight line towards the closest beaconing node, thus 'hitting' a point on the hole boundary. It will then move along the mapped trajectory at a distance approaching the critical communication range, trying to establish a connection with the other sub network of nodes.

Retracing the Routing Path method

This method does not require using either the TENT or BOUNDHOLE algorithms, nor does it require localizing any nodes. Instead, information from the source node to sink node forwards along only one path, called the critical path. All nodes on the critical path store that status of themselves, distinguishing them from non-critical path nodes. When the sink stops receiving information, it floods the network with a message for all nodes on the critical path (and which are still active) to begin beaconing. The mobile robot is deployed, following the beacons of the nodes one by one until it reaches a node where no new beacon is detected, it has reached the hole boundary and Robot moves to a near critical distance from the last static node and traces a circle around it while trying to establish a connection with the other side

Virtual Spring Method

This method also does not require using either the TENT or BOUNDHOLE algorithms, nor does it require localizing any nodes. When the sink stops receiving information, it flood the network with a message for all active nodes to begin beaconing. The Robot will move in a straight line in any direction while listening to beacons. If the beacon signal strength falls under a certain threshold, the robot knows it has passed the boundary. The received signal strength it thought of as a spring or a force between the robot and boundary nodes that the robot wants to keep at an equilibrium. The robot will make progress around the boundary while keeping the spring at an equilibrium and trying to establish communication with the other sub network.

Currently, we are attempting to address the problem using the algorithm that is the easiest to implement (Virtual Spring), not necessarily the most robust. If there is time remaining we plan on increasing the effectiveness of the algorithm by incorporating some of the techniques described in the other algorithms.

Wireless Sensor Network

Technical Description

Overview

To demonstrate the hole detection problem outlined in the introduction and to test the algorithms we will use to remedy it a wireless sensor network system must be created, in our case purchased. A wireless sensor network, at its core, consists of the following components along with their product names:

Base station (MIB510):

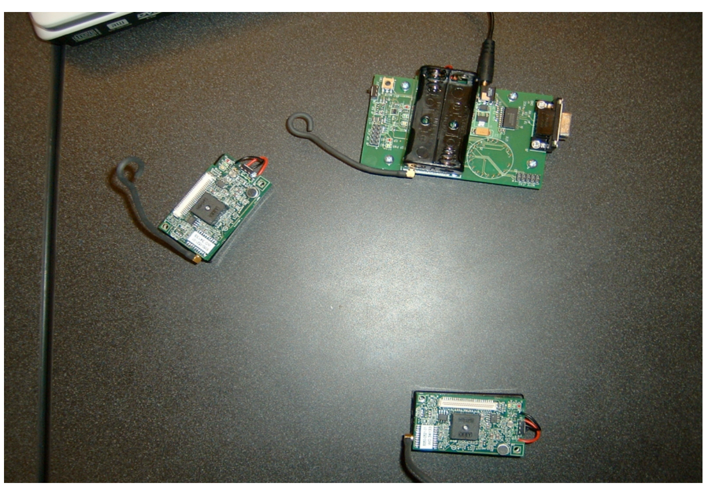
This device enables two way communication from a computer (Or any outside source with computing power) to the wireless sensor network. (The MIB510 currently requires a mote to be attached to it so that it can communicate with the rest of the network)

Mote (MPR 400 / Mica2):

This device consists of a very small microprocessor, some form of internal memory and the ability to transmit wireless messages. In a typical WSN there can be many of these and they form the heart of the network as all information travels across a subset of the motes. The key is this must be small and very power efficient.

Sensor (MTS310):

This device typically connects to a Mote and provides sensory information such as temperature/light/audio that can be relayed to the other motes or the base station.



Crossbow MIB510, MPR400 with MT310 attached

Although the parts above are sold individually by different manufactures they are usually offered in a kit. In the initial phases of the project we intended to purchase a kit from ScatterWeb, unfortunately, around the time of the purchase they stopped manufacturing them so we went with our alternative from Crossbow (See obstacles section).

To create applications that can execute on the Mica2 the memory and power constraints must be taken into consideration. Like regular pc's, where the operating system allows the user to write code without worrying about lower level functionality TinyOS (http://www.tinyos.net/) provides this layer of abstraction for the motes as it is tailored for wireless sensor networks. To effectively write code for TinyOS a programming language that complies with its basic architecture, particularly, its event-driven design should be used. Such a programming language is nesC that is an extension to the C programming language. Although understanding the technical details regarding nesC's functionality are not necessary to explain the project some of our current and possibility future design decisions are made to satisfy certain constraints of the programming language.

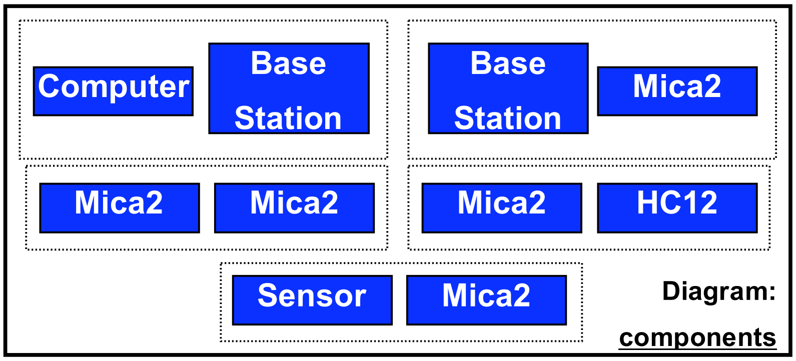
Simulating the Hole Detection Problem

To be able to simulate the hole detection problem we need a concrete scenario we can model. In our previous reports we indicated that we would be transmitting a video feed from the source to the sink across the network. Although we would still like to achieve this we have changed the information type that we will be transmitting (See obstacles section). Our first goal will be to transmit simple temperature/light level information across, then if time permitting we will move on to using a video feed. We have already achieved this.

Currently we are in possession of three mica2 motes, one base station, and two sensors. One of the mica2 motes is attached to the base station another will be needed to interface with the robot (See mica2 to HC12 section) and finally one will have a sensor attached to it. To realistically simulate the problem we will need to, and are currently in the process of, ordering more mica2 motes.

Implementing the Hole Detection & Correction Algorithm

To translate the high level algorithm discussed (See Hole Detection Algorithms section) to a physical implementation the following issues must be addressed (we will deal with them in a direction flowing from the sink to the source see "*Diagram components"*):



Computer ↔ Base Station & Base Station ↔ Network::

To communicate to the base station from a commuter a RS232 serial cable is used. To be able to communicate to the rest of the network the base station must have a mica2 attached to it. In essence the base station acts as bridge between the computer and the rest of the network, it is also transparent to any user. These components are crucial as they facilitate the flow of traffic in and out of the network through the sink. The "*sensor program*" above illustrates *pulling* information from the network to the computer to be shown to the user. Another program that has been created "*push packet*" *pushes* information into the network to direct all the mica2 motes to carry out a particular application (In this case to simply turn on/off the green LED). To implement the hole detection & correction algorithm we will need to direct all the mica2 motes still in range to start broadcasting (i.e. Initiate recovery mode), these components facilitate this process.

Mica2 ↔ Mica2:

To be able to transport information from the source to the sink all the motes along the routing path must forward the packets. This is an important step and is automatically carried out by the underlining routing algorithm installed on the motes. We will also require the mica2 to broadcast signals when in recovery mode so that the robot's mica2 can detect their presence (We are currently implementing this feature).

Sensor ↔ Mica2:

As demonstrated in the "sensor *program*" this component relays the information detected by the sensor to the Mica2 that is then forwarded through the network to the sink. Although this is currently implemented we would like to enable a video feed to be transmitted (See obstacles section for details on how this may be accomplished).

Mica2 ↔ HC12:

Finally, to inform the robot where to move its corresponding Mica2 mote must give it directions. In our solution the robot itself is dumb; it only responds to driving direction (e.g. Move left for 1 second). The calculations that will be carried out on the computer and communicated to the robot through its mica2 using the above methods. Connecting the Mica2 to the HC12 involves:

Whats left to do

Currently, many of the small components necessary to implement the algorithm but there are still parts that need to be devolved:

Broadcasting radio signals from the mica2 (Beaconing)

Transferring information between the mica2 and the robot (I2C communication)

Detecting the signal strength of received packets on the robot's mica2

Relaying the driving instructions to the robot from the computer

Many of the above problems are likely easy to implement and require very little coding. The difficulty lies in finding the write commands and interfaces to use (See obstacles section).

Please refer to this link to view our Gantt chart of the project's current milestones and their expected completion times.

http://www.widar.ca/milestones_updated.html

Bugdet Analysis

Store	Description	Quantity	Price
Hobby Engineering	Texas Instrument SN74410NE H-Bridge	3	$16.64
Dantona Hosting	Website Hosting	1	$30.68
Netfirms	Domain Registration	1	$10.55
Jameco	DC Motors	2	$55.32
Home Depot	Caster Wheel	1	$5.00
Robotshop	Wheel, wheel mounts, and hub	2	$56.64
RobotShop	Power Supply connector	1	$15.31
RobotShop	Shaft Encoder	1	$65.66
Digikey	51 Pin connector for motes	1	$23.40
Radio Shack	Battery	1	$40.19
Home Depot	Bolts, Screws, Nuts		$17.75
Home Depot	Drill Bit, Bolts, Sheet Metal, Plastic Sheets		$44.06
Circuit city	22 Gague Hookup Wire Spool	1	$10.25
		Total	$391.45

Obstacles faced

From the beginning of the project we have had to deal with problems of varying nature. Although many have been relatively minor there have been a few large setbacks. The most significant of these had to do with the wireless network kit. During the project planning stage we came to a conclusion that the best wireless sensor network kit to purchase was the one offered by ScatterWeb (http://www.scatterweb.com/) mainly due to the high level nature of the code required to carry out different commands and the ease of installation. Unfortunately, the day we were going to purchase the kit it was discontinued without a warning. We contacted ScatterWeb to see if a kit could be ordered but they informed us that it would take months. The workaround to this issue was to order the backup kit we had chosen earlier from Crossbow (http://www.xbow.com/Home/HomePage.aspx).

This led into another problem; the Crossbow kit needed to be shipped from San Jose so we were aware that it would probably take a couple of weeks. The process of ordering was carried out by a local sales representative in Hoskin (http://www.hoskin.ca/). The communication between us was rather slow as not all the relevant information regarding the ordering was presented upfront and multiple dialogs were required. On top of this the delivery time, as we regrettably found out, was quite a bit longer then first stated. We initially planed on contacting Ulya regarding the purchase of the ScatterWeb kit on November 7th, after all the delays and issues the Crossbow kit finally arrived on the January 8th. This has been the single major drawback we have faced which has hampered our initial scheduling by about 1 as we originally set aside one month for the delivery of the kit.

Other minor issues we have and are still facing are the following:

1. There is no concrete API stating all the functions of the mica2 mote, because of this a lot of experimentation is necessary.
2. Making sure our robot drives in a straight line.
3. Connecting the mica2 mote to the robot.

Originally we had planed to transfer a video feed between the source and sink, in this report we have indicated that goal is still desirable but it is no longer the basic solution that we will provide. The reason for this set back directly relates to the issue we had ordering our first choice kit. The ScatterWeb kit had a camera that would be attached to the sensor, our current kit from Crossbow does not have this feature. To remedy this we will now instead transfer simple temperature/light data to begin and then moving on to an audio signal. Time premising we will use another “Base Station” to connect to a computer with a camera on it, this will now be our source. We will then packetize the individual frames and send them over the wireless sensor network.

Conclusion

With the construction of the mobile robot completed and initial programming / testing of the WSN motes and base station well underway, our project is approaching closer to the realization of our proposed solution. Communication and data transferring between nodes has been established. Communication between a node and the mobile robot is the next programming milestone to accomplish. Once this is completed, programming the hole detection algorithm into the motes and base station will be the final major task to solve the problem of reconnecting the unlinked sub networks.

After weighing the three proposed search/positioning algorithms, which were the Map Boundary Method, Retracing the Method and Virtual Spring algorithms, in respect to both level of difficulty and robustness in their implementation, the best algorithm to implement was determined to be the Virtual Spring algorithm. This algorithm requires no spatial coordinate information about nodes to determine the location of the communication hole in the network. It exclusively uses signal strength information from static nodes to the mobile node to search for the boundary of the gap. Therefore less information storing and transferring while searching for the network hole is required, as well as fewer calculations regarding positioning. Thus the algorithm can execute faster and more efficiently than the other proposed ones.

With respect to the development of mobile robot aspects of the project, the next phase in the integration of the robot into the project is calibration. This is critical as it directly influences the practical application of the hole detection algorithm. If the robot cannot navigate precisely through the network of nodes, it will not be able to accurately execute the steps of the Virtual Spring algorithm, ultimately resulting in a failure to locate the communication hole and solve the network link problem. Once properly calibrated, the robot will then enter a phase of practical testing of the algorithm encoded into the MICA2 motes, at which phase of the project our team will be able to determine if our proposed solution is feasible.