SCHOOL OF
ELECTRICAL AND
ELECTRONIC ENGINEERING

THE UNIVERSITY of ADELAIDE

Stadium Spectacle
System Simulation

Syeda Mahfuz (a1175283)

ELEC ENG 4039 A/B HONOURS PROJECT

Final Report

Date submitted: 21st October 2011

Supervisors: Dr. Braden Philips, Assoc. Prof Michael Liebelt

Acknowledgments

I would like to thank my project partner Ying Ying Tiong. She has been a wonderful and dedicated team member, and has been an excellent source of motivation for me during the course of the project. Next, I would like to thank the project supervisors, Dr. Braden Philips and Assoc. Professor Michael Liebelt. They have always guided us towards the right direction throughout the project and patiently endured all our questions. Their hints and guidance have enabled us to ask and answer questions that we would otherwise not have considered and thus have made our experience all the more valuable. I would also like to acknowledge Danny Giacomo, always been willing to answer questions that we have had about hardware components and he has also provided guidance on how to research and look for parts. He has also taken care of ordering necessary components for us. Thanks also to Pavel Simcick and all the workshop staff for their time and support. Last but not least, we would like to thank the school of Electrical and Electronic Engineering in the University of Adelaide, for giving us the chance to get involved in the project, thus allowing us to gather experience and knowledge on designing electronic systems and helping us to be independent learners.

Executive Summary

The stadium spectacle project is about developing a prototype of a new system that provides spectacular light shows suitable for an opening ceremony of a sporting event. The system is made up of wireless network of badges that can form ad hoc wireless networks. Each badge has a radio module to communicate and also has LEDs that can flash on and off. A PC base station is also required to tell the badge what type of light effect to create. The badges need to be small so that spectators in an event can attach the badges onto their hat or shirt. The badges also need to be low power and cheap. The system is split into two parts, System Development which constructs the prototype badges and a PC base station, and System Simulation which develops a simulation of the system with hundreds of badges.

We have considered Zigbee, Bluetooth and ANT as the wireless network protocols to be chosen for this project. ANT has been chosen because it is much more lower power, cheaper and smaller.

This report contains details about the System Simulation part of the project which involved designing the communication strategy between the badges that would allow them to create spectacular light effects. Two designs were considered, one that used Auto Shared Channels and the other used simple broadcast networks. Simple broadcast networks were chosen to be implemented for this project due to its simplicity and also because they are more power efficient.

The project has been completed successfully, since 2 prototype badges have been constructed along with a PC base station, all of which can communicate wirelessly to one another and flash LEDs on and off. A simulation with hundreds of node that can create different light effects has also been developed. The final cost of the system has been estimated, and the badges will cost less that \$10 each and 1 PC base station would cost less that \$550.

Contents

1 Introduction 6

1.1 Motivation . . . 6

1.2 Aim . . . 6

1.3 Background and significance . . . 6

1.4 Requirements . . . 7

1.4.1 Final Deliverables . . . 8

2 Study of Wireless Network Protocols 9

2.1 Option 1: ANT . . . 9

2.2 Option 2: Bluetooth . . . 10

2.3 Option 3: ZigBee . . . 10

2.4 Preference . . . 11

3 Essential information about ANT 12

3.1 The ANT Node . . . 12

3.2 The ANT Channel . . . 12

3.3 Channel Configuration . . . 13

3.3.1 Channel Period . . . 14

3.3.2 RF Frequencies . . . 14

3.3.3 Channel Type . . . 14

3.3.4 Network Type . . . 15

3.3.5 Channel ID . . . 15

3.3.6 Some other Parameters . . . 15

3.4 Establishing a Channel . . . 16

3.5 Ant Message Types . . . 16

3.6 More about Channels . . . 17

3.6.1 Independent Channels . . . 17

3.6.2 Shared Channels . . . 17

4 **Implementation of the Ant protocol** **19**

4.1 Analogy: Mexican Wave . . . 19

4.2 Design of PC and Badge Nodes . . . 19

4.3 Design using Auto Shared Channels . . . 22

4.4 Design using Simple Broadcast Networks . . . 26

4.5 Comparing the two designs . . . 30

5 **Resources for the Simulation** **31**

6 **Software Design** **32**

6.1 Overview . . . 32

6.2 Assumptions Made . . . 33

6.3 The GUI . . . 34

6.4 Detailed Software Structure . . . 35

6.5 Challenges Faced . . . 38

6.6 Javadoc . . . 39

7 **Results of Simulation** **40**

8 **Testing** **43**

8.1 Manual Testing . . . 43

8.2 Automated Testing . . . 43

9 **Project Management** **44**

9.1 Allocation of Tasks and Timeline . . . 44

9.2 Control and Monitoring . . . 45

9.3 Risk Management . . . 46

9.4 Budget . . . 47

9.5 Configuration Management . . . 48

10 **Final Cost of System** **49**

11 Reflection on Project **50**

11.1 Strengths and Weaknesses 50

11.2 Other Applications 51

11.3 Social and Cultural Impact 51

11.4 Environmental Considerations 51

12 Conclusion **52**

12.1 Summary of Project Outcomes 52

12.2 Lessons Learnt 52

12.3 Summary of Future Work 52

A Original Timeline **54**

B Original Risk Assessment **54**

1 Introduction

1.1 Motivation

We are all familiar with spectacular light shows in Opening Ceremonies of sporting events, such as the Olympics. Devices such as laser or fireworks are often used to create stunning effects that delight the people looking on. Our idea is to create a new system capable of providing fantastic light effects in such an event, while also making the system low power, cheap and small. The system will comprise of a wireless network of badges, such that they can communicate in an ad hoc fashion. There will be one badge each for every spectator within the stadium and spectators will be able to attach them onto their hat or shirt. The badges will contain radio modules allowing them to engage in wireless communication with other badges. They will also contain LEDs that can flash on and off to create light effects. In addition to the badges, PC base stations will also be required, such that each badge has at least one base station in range inside the stadium. The PC base stations are responsible for sending out initial messages, which the badges can subsequently retransmit around the network to create light patterns.

1.2 Aim

There are two major aims of this project.

The first aim is to create a prototype for the system as described in section 1.1. The prototype system will consist of at least two badges and one PC base station. My project partner, Ying Ying Tiong is responsible for this segment of the project. We refer to this segment of the project as System Development.

The second aim is to develop a simulation of the system to allow us to observe what sort of light patterns might be achieved from such a system. It is essential that the simulation be a faithful simulation and reflect the actual capabilities of the real system. I am responsible to this part of the project. We refer to this segment part of the project as System Simulation.

1.3 Background and significance

This project employs "ad hoc wireless networking" and it is important to understand what this term implies and why it is being used. Ad hoc networking refers to a set of network elements that connect to form a network without the need of planning beforehand. Therefore, ad hoc wireless network means that wireless devices within range of each other can discover and directly communicate with each other without the need of central access points. For this project, it means that the badges can directly pass messages to one another. They do not have to know the location or position of other badges to form a network or to form light patterns. Moreover, the badges can leave and join the network at any time without disrupting the system or affecting the light patterns. For this project, ad hoc wireless networking allows spectators to be able to move around the stadium without disrupting the way the badges communicate.

The advantages of using a wireless ad hoc network are:

- There is no need to build a central access points. This also makes the nodes independent since they are free from any central administration.
- There is no single point of failure within the network and this makes the system flexible, reliable and more robust.
- The system is scalable since nodes can join and leave the network at any time.

The disadvantages however are that:

- The security of the wireless network is low, and hackers can join the network and be able to steal information.
- Throughput of the system might be affected by system loading.
- Large networks might have excessive latency .

Whilst in this project, we are working with wireless ad hoc networks to create something fun, it is significant, because there are many other applications than can be created using the same principles. Such applications include tracking tools and unmanned robots. For more information, please refer to Section 11.2.

1.4 Requirements

The requirements as set out by the supervisors in the project charter are as follows:

1. The price of badges should be in order of dollars per badge.
2. Price of a PC base station must be less than $10000.
3. Badges should contain an ID that is specific for each event so that badges from different events cannot communicate.

4. Badges should be generic such that they can be reconfigured for many different events.
5. Badge can be configured wirelessly.
6. Badges should operate in two modes:
   - **Event Mode:** The badges will be in this mode for the duration of an event. In this mode the badges will be able to pass messages around to create different light effects.
   - **Souvenir Mode:** After the completion of this event, the badges will adopt adopt a 'souvenir mode". During this mode, badges will look out for other badges that were in the same event. If it comes within range of one, it will start flashing as if to say: "I was there in that event too!"
7. Badge and PC base station can form an wireless ad hoc networks.
8. Badges should be able to communicate within a radius of 1m or less so that they can talk to neighbouring nodes.
9. Badge should able send messages to a single neighbouring node in any direction, as well as to broadcast to all neighbours within range.
10. Badge should recognise the task and not perform the same task twice.
11. Badge will react differently and create different light patterns based on the received message.
12. The following light patterns can be created:
    - Random Trace - A random trace of light that proceeds around the stadium
    - Bloom - blooms of coloured light emanating from points in the network.
    - Synchronous Flash - Flashes of light in which all badges are synchronised to flash simultaneously.
13. The LEDs on the badge should be able to flash on and off with different colours.

1.4.1 Final Deliverables

The final deliverables are set out in the project charter are:

1. Four or more prototype badges
2. A PC base station
3. A Simulation with hundreds of nodes that can form the light patterns as described in Requirement 12.

2 Study of Wireless Network Protocols

The first step in the project is to choose a suitable wireless network protocol. Currently, the most popular wireless network protocols in the market are ANT, Bluetooth and ZigBee. It is essential that the hardware modules supporting our chosen protocol be as low power and as small as possible. A simple network protocol and low cost are also advantageous. We studied and compared ANT, Bluetooth and Zigbee and keeping these basic requirements in mind, Zigbee and ANT were deemed to be the most suitable protocols for our project. Zigbee is a well known and proven protocol whereas ANT is quite new. However, ANT seems to satisfy these requirements better than Zigbee. Following is the comparison of the three protocols.

2.1 Option 1: ANT

ANT is designed for applications that require periodic transfer of small amounts of information between up to hundreds of interconnected devices or nodes. It supports a variety of network topologies including point-to-point, star, tree and mesh.

ANT is highly power efficient leading to extended battery life. It is also a low cost solution. A list of commercially available products that use ANT can be found at:
`http://www.thisisant.com/why-ant/proven-products`

ANT-powered nodes can act as both masters or slaves within a network. In other words, they are capable of transmitting and receiving signals, as well as acting as a transceiver to enable messages to be passed on, to other nodes. Moreover, every node is able to determine whether to transmit, based on the activity of nodes near it.

ANT supports three types of messages:

1. **Broadcast:** A one way communication between nodes
2. **Acknowledged:** Receiving node sends an acknowledgment to transmitting node to confirm that message has been received
3. **Burst:** A multi message transmission where the receiving node confirms whether message has been successfully received. If not data is re-sent.

ANT employs the use of "adaptive isochronous network technology" to combat interference. Each message is transmitted in an "interference free time slot" and each message takes less than 150 μs to transmit. This means that one single ANT channel can be split into many timeslots. These scheme is flexible, in that if interference from a nearby node is detected, then the transmission timing is changed. This is advantageous because it allows transmitters to lengthen transmission times to avoid interference, however, there is no overhead where there is no interference.

2.2 Option 2: Bluetooth

Bluetooth is designed for fast data transfer between devices, that is in a PAN. Although Bluetooth is less expensive and more power efficient than Zigbee, its stack size is much larger than that of Zigbee. On the other hand, it is inferior to ANT in all these regards.

Bluetooth has a transmission rate in air of 1000 kbits/s which is about the same as that of ANT, but it only supports peer to peer and star network topologies and its maximum network size is only 7. These are major drawbacks that make Bluetooth unsuitable for this particular project.

2.3 Option 3: ZigBee

ZigBee is wireless technology designed based on IEEE 802.15.4 standard for Low Rate Wireless Personal Area Network (LR-PAN). It was designed for radio-frequency (RF) applications which require low data rate, long battery life and secure networking.

ZigBee is simpler and less expensive and WPAN and Bluetooth. Currently, there are two available ZigBee based network solutions: ZigBee Specification and ZigBee RF4CE specification. The former was designed to support mesh network with thousands of devices, whereas the latter was designed for device to device applications.

ZigBee protocols are built on recent algorithmic research (Ad-hoc On-demand Distance Vector, neuRFon). Thus, the system is able to construct a low-speed ad-hoc network of nodes automatically. There are two type of ZigBee network protocols modes: beacon and non-beacon enabled. In a beacon network, devices switch to sleeping mode automatically when not transmitting, whereas in non-beacon enabled network, some device are always active while others only active while transmitting.

ZigBee networka are composed of three device types: ZigBee Coordinator, ZigBee Router, and ZigBee End Device. The coordinator controls the formation and security of the network, the router extends the network, and the end device senses or controls the devices.

2.4 Preference

Table 1 summarises the key features of the three network protocols.

Table 1: Key Features of Optional Wireless Protocol

	ANT	**ZigBee**	**Bluetooth**
Standard	Proprietary	IEEE 802.15.4	IEEE802.15.1
Application	PANs and WSNs	PANs and WSNs	PANs
Host resources (kBytes)	2 (0 with SensRore)	100	250
Battery life (with coin-cell battery)	3+ years	4 to 6 months	1 to 7 days
Max. network size (nodes)	2^{32}	2^{64}	7
Over the air transmission rate (kbit/s)	1000	250	1000
Required PCB area (mm^2)	125	Depends	Depends
Range (m)	1 to 30	1 to 100+	1 to 10+
Success metrics	Ultra-low power, cost	Power, cost	Cost, convenience
Supported networks:	Peer-to-peer, star, tree, mesh	Peer-to-peer, star, tree, mesh	Peer-to-peer, star
Min. node configuration:	Transmit or transceiver	Transceiver	Transceiver

As has already been mentioned, Bluetooth is not a suitable choice for this project. So, lets consider the advantages and disadvantages of ANT and Zigbee which are quite similar. Both support the same network topologies and aim to be power saving and easy to use. However, ANT is significantly more power saving and provides a lower cost solution. In fact, ANT is 2 times more power saving than Zigbee and 60 percent less BOM cost. Ant is also has a much simpler protocol to work with, reducing development time. ANT's stack size is 10 times smaller than that of Zigbee. As can be seen from Table 1, ANT has a faster transmission rate that Zigbee in air.

Although Zigbee provides a larger network size, the maximum network sizer provided by ANT is more than sufficient for this project. Also, Zigbee is a proven protocol, whereas ANT is relatively new, being only about 6 years old. However, the enhanced power saving capabilities, smaller size and lower cost all are significant advantages that result in us choosing ANT as the preferred wireless network protocol for this project.

3 Essential information about ANT

3.1 The ANT Node

ANT networks are composed of nodes, all of which have an ANT engine and a host microcontroller. The ANT engine we have chosen to purchase is nRF24AP2 since this is the newest one and hence, has more functionality. Figure 1 depicts an ANT node.

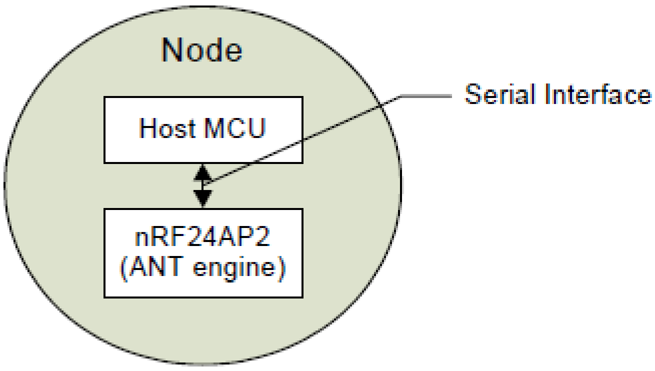


Figure 1: An ANT node

The ANT engine is responsible for connecting to other ANT devices whereas the host microcontroller's task is to implement the functionality of the device.

3.2 The ANT Channel

For communication between ANT nodes, a channel must exist between the communicating nodes. Each channel must have at least one master node and one slave node. Slave nodes search for master nodes and whenever it finds a master it connects to it. The master node always initiates the communication by broadcasting messages, allowing slave nodes to detect and connect to it. The master node is the primary transmitter, however it is also able to receive information from the slave node, such as confirmation of whether the slave node received the message or not. Similarly, the slave node is the primary receiver, however it can still transmit information back to the master node if necessary. A message passed from the master to a slave is said to be in the forward direction, whereas the opposite is said to be in the reverse direction.

An ANT node is able to open up to eight channels simultaneously. The node can be a master node or transmitter on some channels and it can act as a slave node or receiver on another channel. Effectively, this means that ANT nodes can act as transceivers and they can receive messages from one node and pass messages onto other nodes in the network.

The master node sets the timing of a channel, that is, the interval at which messages are sent. The "slave locks onto the timing set" in order to receive messages from the master. The slave can send a message back if it is configured to do so. The slave can send an acknowledgment or data as appropriate. Figure 2 shows the communication between a master and a node.

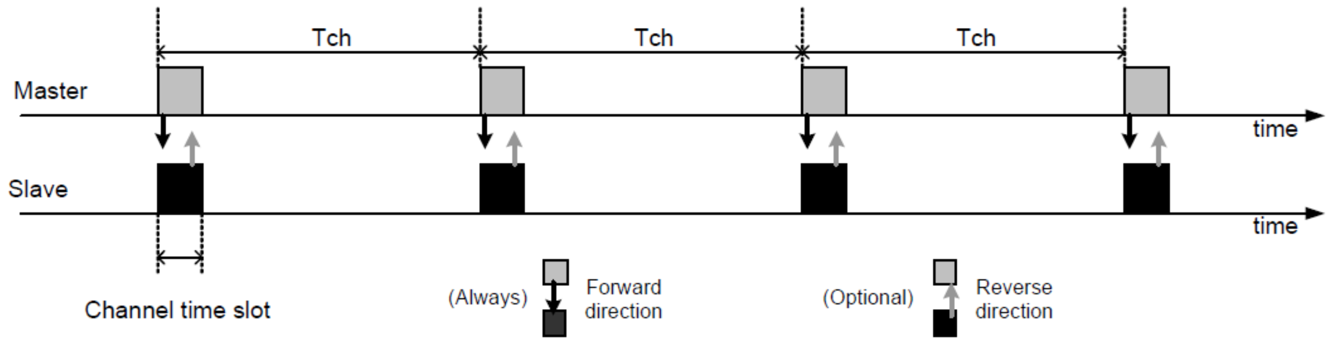


Figure 2: Communication across a Channel

On each channel timeslot, up to 8 bytes of data can be transferred, in both directions. The data can be send as either, broadcasts, acknowledgments or bursts. For more information on ANT message types, please see section 3.5.

Every channel time slot the master sends a message across the channel. If there is no new data, then the last packet of information is sent.

The most simple ANT channel is called an independent channel. It consists of just one master node and one slave node.

3.3 Channel Configuration

To set up a channel, the nodes in the channel must have the same channel configuration. The parameters that need to be set are shown in Figure 3. The channel configuration parameters are constant throughout the system. The channel ID is specific to each master node in the network and it is used to identify one master node to another. Slave nodes can use the channel ID to find and connect to specific master nodes.

Let's now take a closer look at each of these parameters.

Parameter	Comment
Channel configuration	
Channel period	Time interval between data exchanges on this channel (5.2 ms - 2 s)
RF frequencies	Which of the 78 available RF frequencies is used by this channel
Channel type	Bi-directional slave, bi-directional master, shared bi-directional slave, Slave Receive only
Network type	Decides if this ANT channel is going to be generally accessible (public) to all ANT nodes, or if it shall limit its connectivity to devices belonging to a managed or private network
Channel ID	
Transmission type	1 byte – Identifying characteristics of the transmission, can for instance contain codes on how payload is to be interpreted
Device type	1 byte - ID to identify the device type of the channel master (Ex: heart rate belt, temperature sensor etc.)
Device number	2 byte - Unique ID for this channel

Figure 3: Configuration parameters

3.3.1 Channel Period

The channel period is the rate at which data is sent by the master node. It is a 16-bit number and is given by:

$$ChannelPeriodValue = \frac{32768}{MessageRate(Hz)}$$

By default, the message rate is 4Hz. A higher message rate leads to increased power consumption.

3.3.2 RF Frequencies

This is the frequency on which an ANT channel operates. An ANT channel may operate on any frequency within the range 2400MHz to 2524 MHz, separated by 1 MHz. Thus ANT channels can operate on 78 different RF Frequencies. The default frequency is at 2466MHz.

3.3.3 Channel Type

The channel type is a 8-bit number (between 0 and 255) and it indicates the type of communication that will take place on the channel. Some of the most common channel types are:

1. **Transmit Only** - data can only flow in the forward direction and hence the master node cannot receive any information.

2. **Receive Only** - data can only flow in the forward direction and hence the slave node cannot send any information.

3. **Bidirectional Channel** - the data can flow in the both the forward and reverse direction. There are two modes, master and slave, which indicates the primary direction of data flow. For example, a bidirectional slave channel means that the slave will primarily receive, however it can still transmit in the opposite direction.

4. **Shared Bidirectional Channel** - many nodes share an independent channel with one central node which receives and processes data from all other nodes in the channel.

3.3.4 Network Type

There are three types of networks, public, managed and private. A public network means all the nodes are publicly available. A managed network can be used when the nodes must comply with certain rules and regulations. A private network is used to restrict access to the nodes. For nodes to communicate, they must belong to the same network. By default, the network is set to public.

3.3.5 Channel ID

There are three parts to the Channel ID:

1. Transmission Type - contains information on the transmission characteristics of the device

2. Device Type (+ Pairing Bit) - used to distinguish different types of nodes within a network.

3. Device Number - a unique number to identify each single node within a network

3.3.6 Some other Parameters

A master node needs to set one more parameter before it can open a channel, namely the transmission power. This value ranges from 0 to 4, such that:

- 0 - power of -20dBm

- 1 - power of -10dBm

- 2 - power of -5dBm
- 3 - power of 0dBm
- 4 - power of 4dBm

By default, the transmission power is set to 3 (0 dBm). A higher transmission power will result in higher current consumption, as well as affecting the area on which the ANT device can influence. We conducted an experiment to determine how the transmission range is affected as the power level is changed. We discovered, that the transmission range increases as the power level is increased, and also that at the higher power level, the transmission range is around 30m which agrees with the ANT documentation.

A slave node needs to set an additional timeout parameter which specifies how long the slave node should search for a master node. If no master node is found within the timeout period then the slave node will close its channel.

3.4 Establishing a Channel

For two nodes to communicate, the channel IDs should match, however, it is not necessary to have prior knowledge of the channel ID. In ANT, slave nodes find and connect to the master node. Hence, all of the fields of the channel ID must be set on the master node. If a slave node knows the exact channel ID of the master it wants to connect to, then it can set all the fields of its channel ID to that of the master it is seeking in order to find the master. Slave nodes can also connect to unknown masters by setting one or more of the channel ID fields to zero (wildcard).

A master node broadcasts it channel ID when it opens a channel. Once the channel is open, the master will continually sent information at the message rate, until the channel is closed.

3.5 Ant Message Types

ANT supports three types of messages:

1. **Broadcast:** This is a one way communication between nodes. There is no acknowledgment that data has been received or not. This type of communication takes up the least number of RF Bandwidth and consumes less power. If occasional data loss is tolerable, then broadcast messaging is a good option. Data can be sent in the reverse direction if specifically requested.

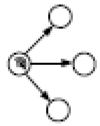


Figure 4: A broadcast network comprised of independent channels

Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7

Figure 5: Data Payload of Independent Channel

2. **Acknowledged:** The receiving node sends an acknowledgment to the transmitting node to confirm that the message has been received. This takes up more RF Bandwidth as well as power, compared to broadcast messaging, however it ensures that communicating nodes are aware of whether data has been received or not.

3. **Burst:** This is a multi message transmission, where the receiving node confirms whether a message has been successfully received. If not data is re-sent.

3.6 More about Channels

3.6.1 Independent Channels

An independent channel has only one master and one slave. However, it is possible for the master or slave node to be a master or slave node to many other nodes.

A broadcast network can be composed of such independent channels as shown in figure 4. In this case data flows in one direction - from the master to the slaves.

ANT has 8 byte data payloads. The data payload of an independent channel has the structure as shown in Figure 5.

3.6.2 Shared Channels

Shared channels are when multiple nodes share one independent channel to communicate with a central master node. For shared channels, the first two bytes of the data payload define the shared channel address. Each node within the shared channel is assigned a different shared channel address. This means that, up to $2^{16} = 65536$ nodes can be on a single shared channel.

Shared Address LSB	Shared Address MSB	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5

Figure 6: Data Payload of Shared Channel

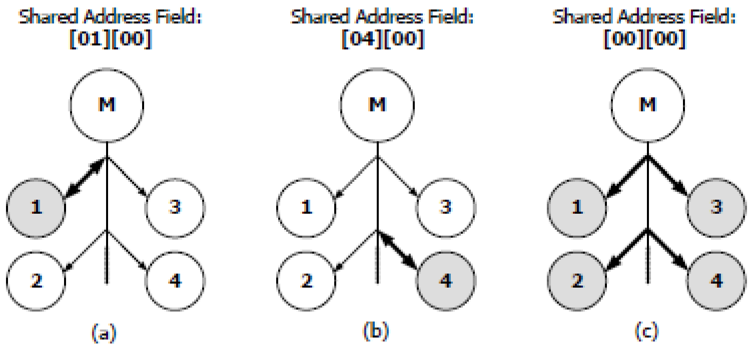


Figure 7: A bidirectional shared channel

In this case, the master node can send information to all slave nodes by a setting the shared channel address to 0. It can communicate to a specific node by setting the shared channel address to that of the node it wants to communicate to. The slaves cannot all communicate to the master at once. However, if the master speaks to a specific slave node, then the slave node can send data back to the master. Figure 8 depicts the behaviour of a shared channel.

A special type of shared channel is the “Auto Shared Channel”. In this case, nodes can form an ad hoc shared network and nodes can join and leave the network at any time. Figure 8 shows an Auto Shared Channel.

The use of shared channels increases the power consumption. It also reduces the the number of useful data types to 6 (as compared to 8 for independent channels). However, the latency is the same as that of independent channels.

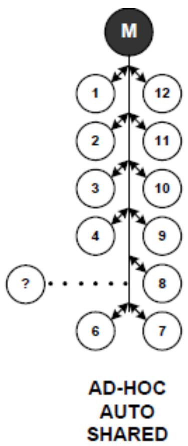


Figure 8: An auto shared channel

4 Implementation of the Ant protocol

This section deals with how the ANT Protocol can be implemented to form ad hoc networks and to create the light patterns. Two designs are considered, one uses Auto Shared Channels and the other uses simple broadcast networks. However, before delving into these details, it is useful to understand at a higher level how we want the nodes to behave and how each individual node is going to be configured.

4.1 Analogy: Mexican Wave

In our system, we have two types of nodes:

- PC nodes - these nodes are the base stations. They are responsible for determining which light pattern needs to be performed, based on user input, creating the appropriate message for each light pattern and sending out this message to other nodes.
- Badge nodes - these nodes are the badges which each spectator has. They are able to receive and transmit messages and flash LEDs on and off to create light effects.

A good way of understanding how the nodes communicate is to think of a Mexican wave. Imagine that there is a room full of people and one person decides to do a Mexican wave. The person whispers quietly, Everyone do a Mexican wave and Pass it on. Everyone who hears this person does a Mexican wave and they also whisper the message and this process continues until everyone has done a Mexican wave.

The badge nodes behave in very much the same way as shown in Figure **??**. The PC Node can be thought of as the person who decides to start the Mexican wave. The PC broadcasts the first message and all within range of it receive it. They carry out the instructions and if required, they pass the message onto other nodes that are within range of them.

4.2 Design of PC and Badge Nodes

As has previously been mentioned, PC nodes only need to create and send out the initial message for a given pattern. Hence, PC nodes need to act as transmitters only. Consequently, only one channel, set as "Transmit" needs to be assigned and opened for the PC node.

In contrast, the badge nodes not only need to receive and decode messages, they also need to be able to pass messages on. They need to act as *both* transmitters and receivers, also

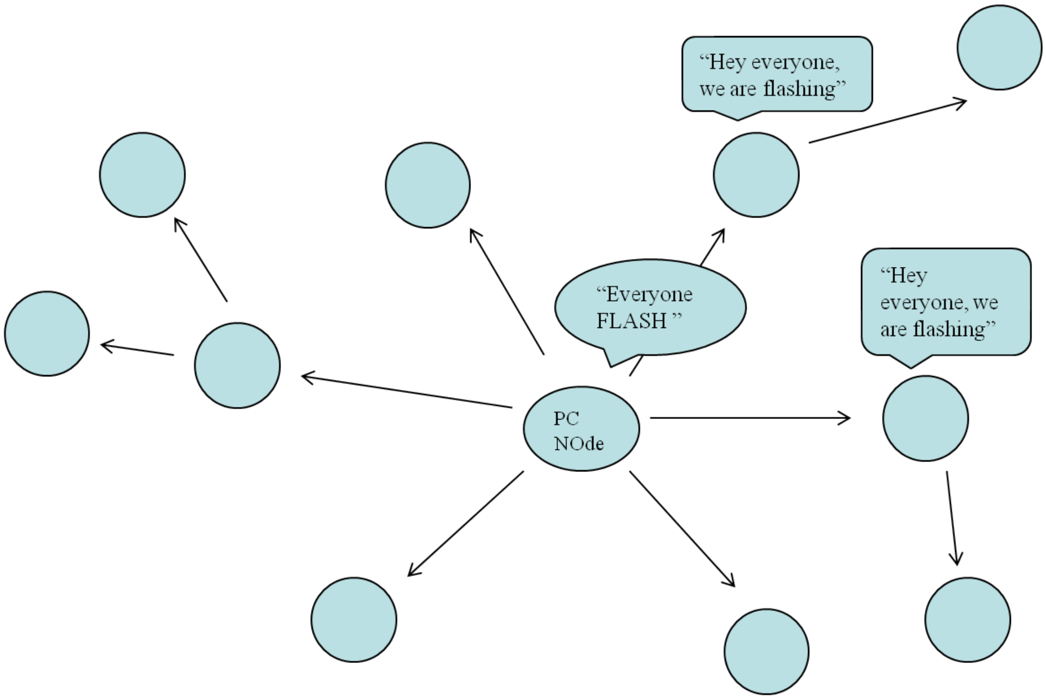


Figure 9: Nodes communicating

known as transceivers. To achieve this, badge nodes must use two channels, one as a Receive channel and the other as a Transmit channel.

The Receive channel must be kept open at all times because the badge nodes continuously need to listen in for master nodes, so that it can connect to the master node in order to receive messages and take part in the light effects. The transmit channel however, need only be opened if the badge node is required to pass a message on. As has been mentioned in Section 3.2, whenever a transmit channel is open, it will keep on broadcasting data at every period until the channel is closed and this is not very power efficient. So, we have decided to only open the transmit channel if the badge needs to retransmit a message, and as soon as the message has been retransmitted, the transmit channel will again be closed.

Note that this means that all the badges in the stadium will *not* be connected up at all times. Whenever, a PC node wants to start a light pattern, it will open its transmit channel, and badges in range of the PC will be connected to it. However, badges outside the range of the PC node will only connect to another node if another badge node within range opens its transmit channel. Figures 10 and 11 helps to understand how the network will work.

In Figure 10 the yellow nodes represent the nodes that have established a connection with the PC node, whereas the blue nodes are not connected to anything. They just have their receive channels open and are listening in for a transmit channel to open in a node that is within range of them so that they can connect to it.

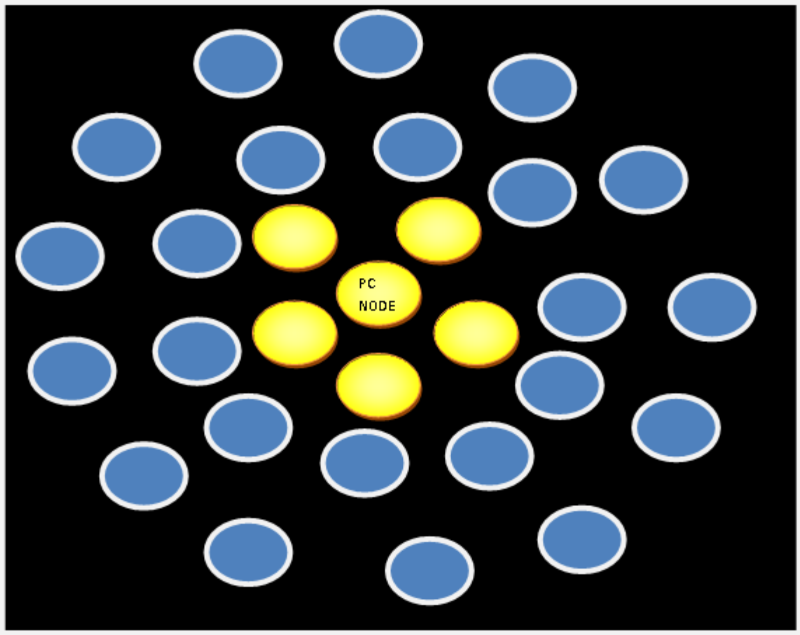


Figure 10: Only PC node has transmit channel open

Figure 11 represents the situation when another node has also opened its transmit channel. Let's call this node NODE 1. Here, we find that the nodes in range of the PC are still connected to the PC (they are coloured yellow). The nodes in range of the NODE 1 are now able to connect to NODE 1, and they are coloured pink. Note that NODE 1 was one of the badges that was connected to the PC in Figure 10. In this situation,it is still connected to the PC as a receiving node on one channel channel. But it is open as a transmitter on another channel and hence, it is acting as a transceiver at this point.

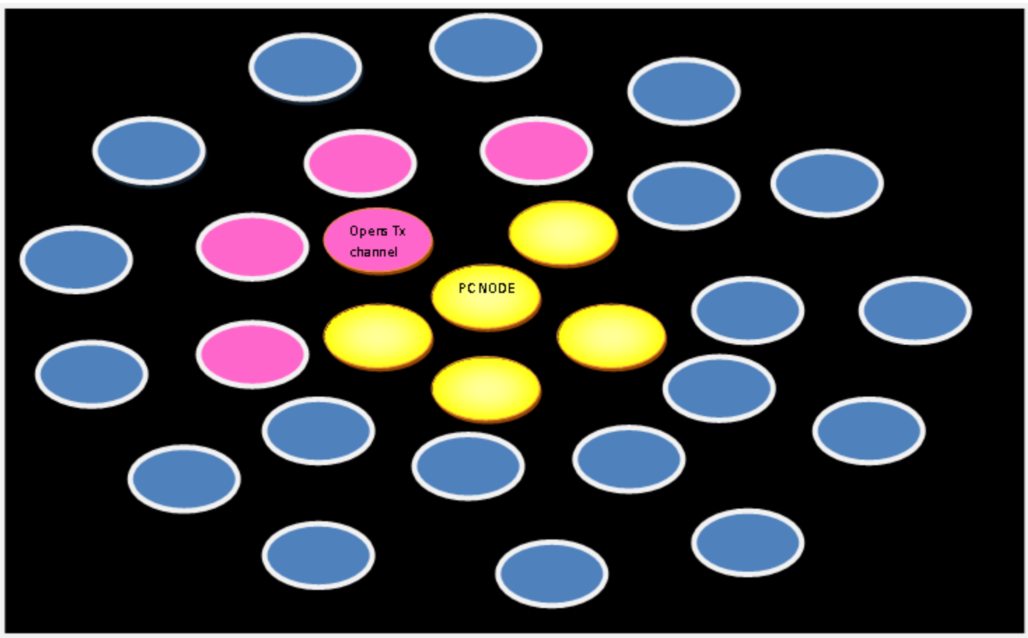


Figure 11: Another node opens a transmit channel

4.3 Design using Auto Shared Channels

In my first attempt at designing our system I looked at using Auto Shared Channels. This section looks at the communication strategy required to achieve the different light patterns using Auto Shared Channels as well as the required message structure.

The Communication Strategy to achieve light patterns

Random Trace

- PC selects any random node around it
- The PC says to the selected node “Flash and Pass On”.
- The selected node flashes
- The selected node now picks out another random node and tells it to “Flash and Pass On”.
- The process repeats until a certain number of flashes have been reached.

Bloom

- PC node selects a random node
- PC tells the randomly selected node “We are doing a bloom starting with you”.
- The selected node flashes and tells the nodes around it, “Everyone Flash”.
- The nodes flash
- The nodes that had flashed now tell nodes around them “Everyone Flash”.
- If a node has already flashed, then it will not do so again.
- The process repeats until all nodes have flashed once

Synchronous Flash

- PC node sends out a message saying “Everyone Flash now”.

A Message Structure

A suitable message structure needs to be devised to ask nodes to behave as outlined above. Using shared channels means that there are 6 bytes of data available to us. The message will contain the following fields:

- **Event ID** - each event will have a specific event ID to prevent badges from different events communicating to each other.
- **Task ID** - whenever the PC starts of a light pattern, the message will be given a task ID. When this message is passed around the network, the nodes will be able to see whether they have performed a task with this ID before. If they have, then they will not respond to the message again. This is to ensure that for a pattern, the lights do not just flash in one spot, but move out from the point of origin.
- **LED Colour** - The colour which the LED should flash.
- **Type Of Pattern** - This will notify the badge whether the pattern is Bloom, Random Trace or Synchronous Flash. Having this information will allow the Badge to determine whether to retransmit the message or not. The badge will also be able to determine whether to pick out and retransmit to just one node (as in Random Trace) or to all surrounding nodes (as in the Bloom).
- **Flash Duration** - This will tell the badges how long the LEDs should be turned on for in milliseconds
- **Number Of Hops** - This will tell the badges the number of times this particular message needs to retransmitted, which we refer to as the number of hops.
- **Current Hop Number** - This will tell the badge the current hop number and this number will be updated every time a message is retransmitted.

A High Level Design to Form the Light Patterns

Figure 12 captures the high level design of forming light effects using Auto Shared Channels.

From the design it is evident that the synchronous flash is very simple. It only requires the PC node opening a Auto Shared Channel as the transmitter. It then transmits the message at the highest power level. Of course, in a real stadium, one PC node will not reach all the badges. Hence, in that situation, multiple PC nodes will be required,one within range of each badge, and they will all need to open Auto Shared Channels and broadcast the message simultaneously so that all badges get the message at the same time and flash together. To achieve this, all of the PC base stations are required to be connected to each other by cables.

The random trace and bloom are more complex to achieve since they both involve multiple numbers of hops and in each hop, the master node needs to change to a new node or set of nodes within the network. For both cases, the transmission power level needs to be set at the lowest level. This is because for the random trace, we want to see lights gracefully moving away from the point of origin (i.e. the PC node), and not jump all over the place in the network. Similarly, for the bloom, we want too observe an effect where lights gracefully spread out from the point of origin. If the transmission power level is set to the highest, there is a danger that the effect might look like a synchronous flash, and so to avoid this possibility, we choose to operate at the lowest power level. Also, in both cases, the nodes need to keep track of the Task ID. Each time the PC generates a new message, it will assign a new task ID to the message. The message will be retransmitted around the network for that particular light patten with the same Task ID as generated by the PC. Badge nodes will keep track if the task ID of the last message that it received. Now, if a node receives a message, it simply has to compare the task ID it has previously saved, and the task ID as appears in the message to determine whether or not it has already taken part in that particular pattern. If it has, it will ignore the message, otherwise it will take part in the light pattern and update its Task ID.

For the random trace, firstly the PC generates the message and then it opens an Auto Shared Channel. Each slave node in the shared channel is assigned a shared channel address from 1 to the number of nodes connected through the shared channel. The PC needs to randomly pick out one of the nodes that are connected to it. To achieve this, it simply generates a random number between 0 and the number of nodes connected to it. It can then use the generated number to address one of the nodes it its network. The PC then closes its channel and the badge node that it had picked out, becomes the new master node. The new master node now opens an Auto Shared Channel and it performs the same tasks as the PC node, except for generating the message. This process continues until the required number of hops have been reached.

The bloom follows a very similar design to that of the random trace. The difference is that in the random trace, the master node picks out one node to flash and for passing the message on. However, for the bloom, the master node just broadcasts to all nodes in its network and all of these nodes flash and pass the message on. Therefore, for the random trace, there is only one master node in the network at any one point, but in the bloom there may be a set of master nodes.

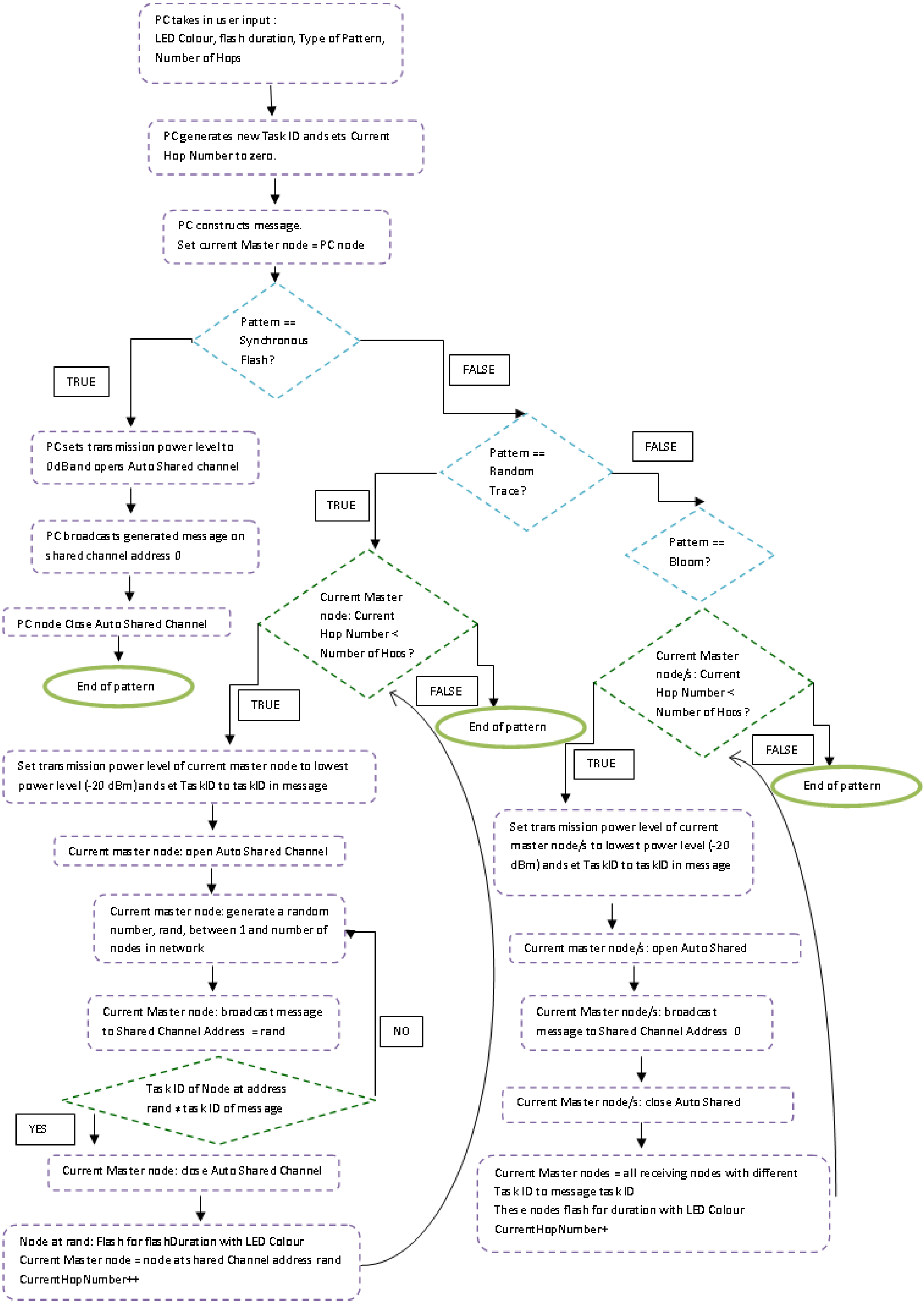


Figure 12: High Level Design Using Auto Shared Channels

4.4 Design using Simple Broadcast Networks

I also looked at designing our system using simple broadcast networks. In this case, the communication strategy is different and it is given below.

Synchronous Flash

- PC node sends out a message saying "Everyone Flash now".

Random Trace and Bloom

- PC broadcasts to surrounding nodes a message saying "Flash and choose if you want to pass on with probability p"
- The nodes Flash and generate a random number between 0 and 100
- If the random number is less than p, the node passes the message on
- The process repeats until a certain number of hops have been reached

In this case, the random trace is different from the bloom by choice of the probability p. In the case of a bloom, p is set to 100, so that all of the nodes pass the message on. However, in the case of a random trace, p is set to a much lower number.

A Message Structure

The message structure used for Broadcast Networks, as shown below, was devised together with Ying Ying. Some of fields are the same as before, however, there are also some differences.

- **Event ID** - each event will have a specific event ID to prevent badges from different events communicating to each other.
- **Interaction Type** - this notifies the badge of what type of message is to be expected. The interaction type can be set to one of the following:
  - PARAMETER - this signifies that the badge is in an event and it will receive a set of parameters that it needs to decode and then perform the necessary task
  - SOUVENIR - to notify the badge to operate in souvenir mode
  - REST - to tell the badge to stop doing everything and turn LEDs off
  - FOUND - Badge only response to this instruction during souvenir mode. LEDs flash when badges receive this instruction.
  - DUMMY - Do nothing

- **Task ID** - whenever the PC starts of a light pattern, the message will be given a task ID. When this message is passed around the network, the nodes will be able to see whether they have performed a task with this ID before. If they have, then they will not respond to the message again. This is to ensure that for a pattern, the lights do not just flash in one spot, but move out from the point of origin.
- **LED Colour** - The colour which the LED should flash.
- **Retransmit Probability** - This will inform the badge of the probability that the badge will retransmit the message. The badge will just need to generate a random number from 1 to a 100 and retransmit if it generates a number that is less than the probability.
- **Flash Duration** - This will tell the badges how long the LEDs should be turned on for in milliseconds
- **Number Of Hops Remaining** - This will tell the badges how many more times the message needs to be retransmitted
- **Power Level** - This will tell the badge the power level at which to transmit messages

A High Level Design of Forming the Light Patterns

Figure 13 displays the high level design in forming light patterns using simple broadcast networks. Note that in this case the flow diagram is constructed from the perspective of any badge node waiting to find and connect to a master node.

For each other light patterns the PC will assign values to each other parameters in the message structure as follows:

1. Synchronous Flash:
    - EventID - would be set at the start of the event and will remain constant throughout
    - TaskID - generate a new value
    - Interaction Type - parameter
    - LED colour - chosen by user
    - Flash duration - chosen by user
    - Retransmit (probability) not required
    - Power level 0dBm (highest)
    - Number of Hops Remaining 0

2. Bloom:

    - EventID - would be set at the start of the event and will remain constant throughout
    - TaskID - generate a new value
    - Interaction Type - parameter
    - LED colour - chosen by user
    - Flash duration - chosen by user
    - Retransmit (probability) 100 - want all badges to pass message on
    - Power level -20dBm (lowest)
    - Number of Hops Remaining chosen by user depending on how big a bloom they want

3. Random Trace:

    - EventID - would be set at the start of the event and will remain constant throughout
    - TaskID - generate a new value
    - Interaction Type - parameter
    - LED colour - chosen by user
    - Flash duration - chosen by user
    - Retransmit (probability) between 5 and 20
    - Power level -20dBm (lowest)
    - Number of Hops Remaining chosen by user

In this again, multiple PC nodes will be required to broadcast the message for synchronous flash together to get all the nodes in the stadium to flash at the same time. Similarly, the different transmission power levels for the different light patterns and the importance of the task ID as discussed in the Auto Shared Channel design still holds in this design.

Please note that in my design I have not covered the souvenir mode. Please refer to Ying Ying Tiong's report for details on the souvenir mode and its implementation.

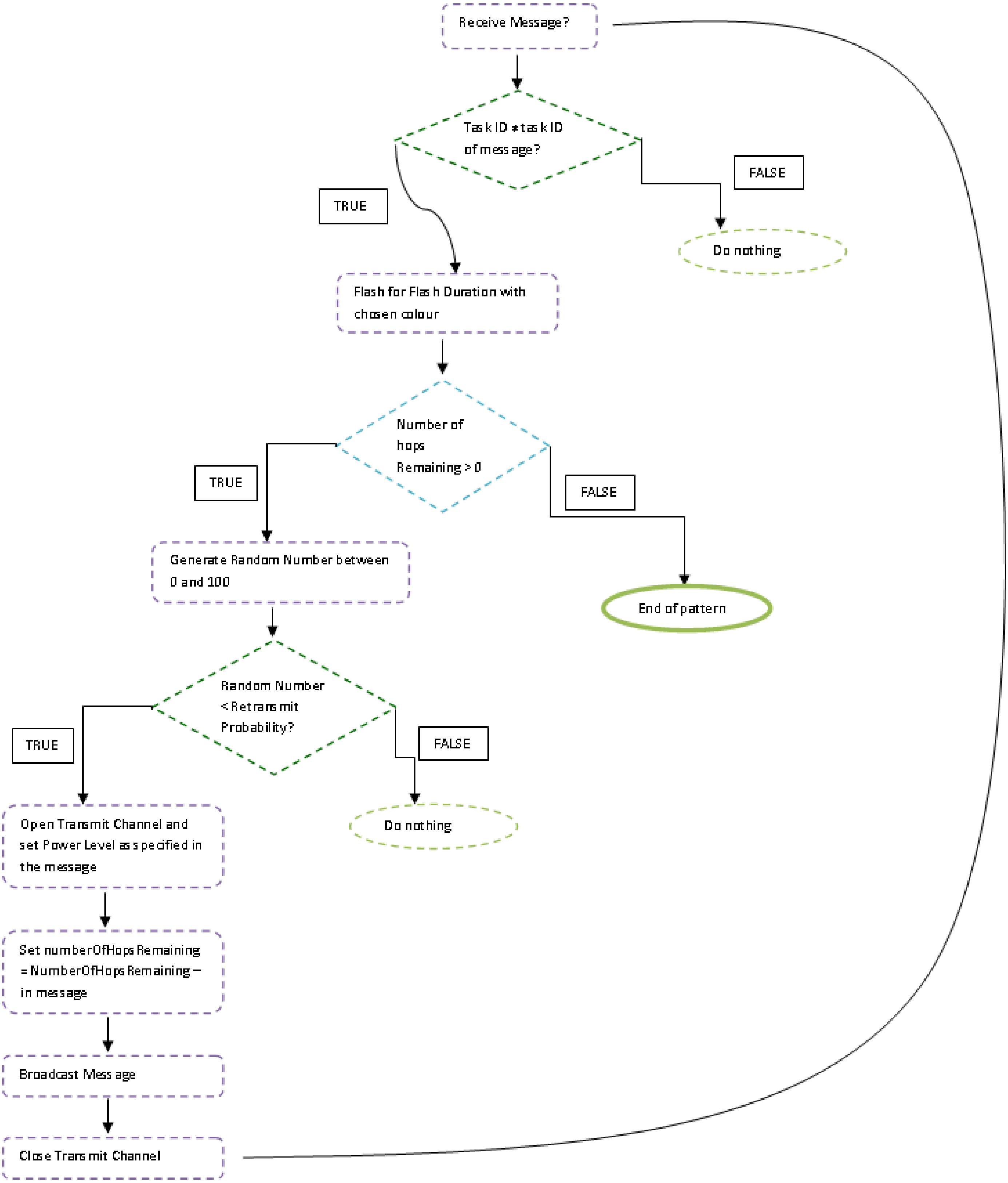


Figure 13: High Level Design Using Broadcast Networks

4.5 Comparing the two designs

In our implementation of the ANT Protocol to form the light patterns, we have decided to use the simple broadcast networks and its associated message structure.

The strength of the broadcast networks and its message structure lies in its simplicity. Different light effects can be created simply by manipulating the message parameters. The message structure is independent of which light pattern is to be done. Badge nodes do not have to know which light pattern in being done at any time with this structure. They simply receive and act as told in the message. The disadvantage, is that only simple light effects can be created with such a simple message structure. More complex effects would require more complexity.

Broadcast networks also have the advantage of being more power efficient and they can utilise up to 8 bytes of data, whereas Auto Shared Channels can only use 6 bytes. Another disadvantage of the Auto Shared channel is that it is more difficult to set up Auto Shared Channels, because it requires not only slave nodes having to find the master node, but the slave nodes also have to be assigned a shared channel address, and it has to be acknowledged on both the master and the slave side that a connection has been successfully established. However, Auto Shared Channels make it easier to send a message to a large number of nodes simultaneously and also, the master node is able to talk to specific nodes within the network simply by addressing its shared channel address. This is not possible in the simple broadcast network. From the flow diagrams in Figures 12 and 13 it can be seen that the design using the Auto Shared Channel is more complex.The added complexity can potentially enable the design using the Auto Shared channels to create more complex light effects.

We have chosen to work with Broadcast Networks only for this project, in order to keep things simple as well as considering the benefits of lower power consumption. However, in future, Auto Shared Channels can be considered to create more complex light effects.

5 Resources for the Simulation

The resources used for the Simulation are:

- **Java** - as the programming language for developing the software
- **Netbeans IDE 6.7** - to develop the software in

Java was chosen simply because of familiarity with Java. Netbeans was chosen because of its good UML support, which helped to model the structure of the software.

6 Software Design

6.1 Overview

The software was split into two different packages:

- **The Ant Interface Package** - the model of the Ant protocol is developed here
- **The Stadium Spectacle Package** - All the graphics is performed here

The class which links the two packages together is called the *RFActivity class*.

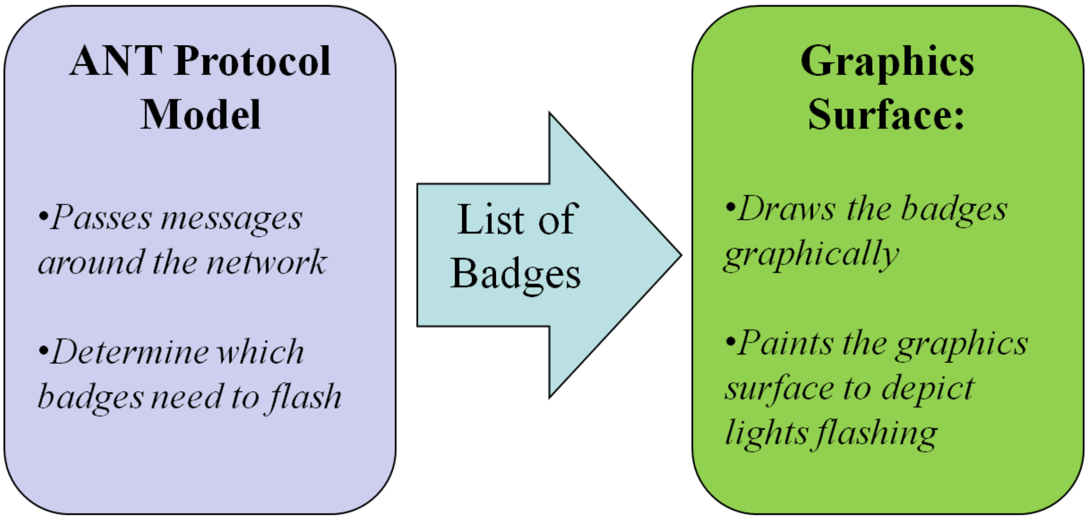


Figure 14: Software Structure Overview of the Simulation

6.2 Assumptions Made

The following assumptions were made in developing the simulation:

1. **Badges are separated by 1 metre in the stadium**
   We assume that spectators are sitting about 1 metre away from each other and hence, the badges are separated by 1 metre.

2. **The power levels correspond to the following transmission ranges:**
   - 0dBm: 20 to 30 metres
   - -5dBm: 15 to 20 metres
   - -10dBm: 5 to 15 metres
   - -20dBm: 3 to 5 metres

Ying and I tried an experiment to try and establish some transmission ranges that each Power Level would correspond to. Basically, we set up the PC node as the transmitter and another node as the slave, and at each power level, we tried to determine the range at which the slave node can still receive information from the PC node. The result of the experiment however, was not very satisfactory and we found that it was not possible to arrive at a certain range of set of ranges that each power level corresponds to. Having said that, we did ascertain that as the power level increases the transmission range increases. We also found that badges in the closed environment with lots of people around achieve lower transmission ranges compared to at an open field with not many people about. I took this fact to assume that in a stadium packed with people, the transmitted radio waves will attenuate very quickly. Also, ANT specifications say that the ANT nodes can transmit a maximum range of 30 metres. Using all of this information, I assumed the ranges as stated in Assumption 2.

6.3 The GUI

The GUI is required to provide an interface between the user and the simulation. The GUI hold the graphics surface on which graphical representations of badges are painted to depicts light flashing on and off. The GUI also contains the buttons that allow users to interact with the simulation.

Figure 15 shows the layout of the GUI. The large grey region represents the stands where the badges are to be placed. The buttons are located at the bottom of the GUI. The buttons have been located in this manner to maximise the stand space and try and fit in as many badges as possible in the simulation. The purple region is used to print out useful information, such as which badges are flashing or which badges are going to retransmit messages.

Figure 15: Layout of the Graphical User Interface

6.4 Detailed Software Structure

The UML class diagram for the AntInterface package is shown in 16. This package simply modeled the ANT Protocol by implementing the available ANT messages which can be found in page 40 and 41 of [13].

The UML Class Diagram of the GUI and Graphics Package is shown in 17. This package created the graphical representations of the badges. These are simply small circles which can be painted with different colours.

The RFActivity class act as the intermediary class between the two packages. This class effectively models the behaviour of Electromagnetic Waves. It contains a list of all the badges and their location on the graphics surface. So, whenever a node opens a transmit channel, the RF Activity class looks at the location of the transmitting node. Then, it looks at the transmission power level of the node and using these two pieces of information it determines which badges around that transmitting node should receive messages that are sent. Moreover, for a given light pattern, the RFActivity class determines which badges are meant to flash and when and it stores this information in an ArrayList. Once the message passing for a given pattern is completed, the RFActivity class passes the list of badges that are meant to flash to the graphics surface which then paints the graphics surface the correct number of times to depict lights flashing.

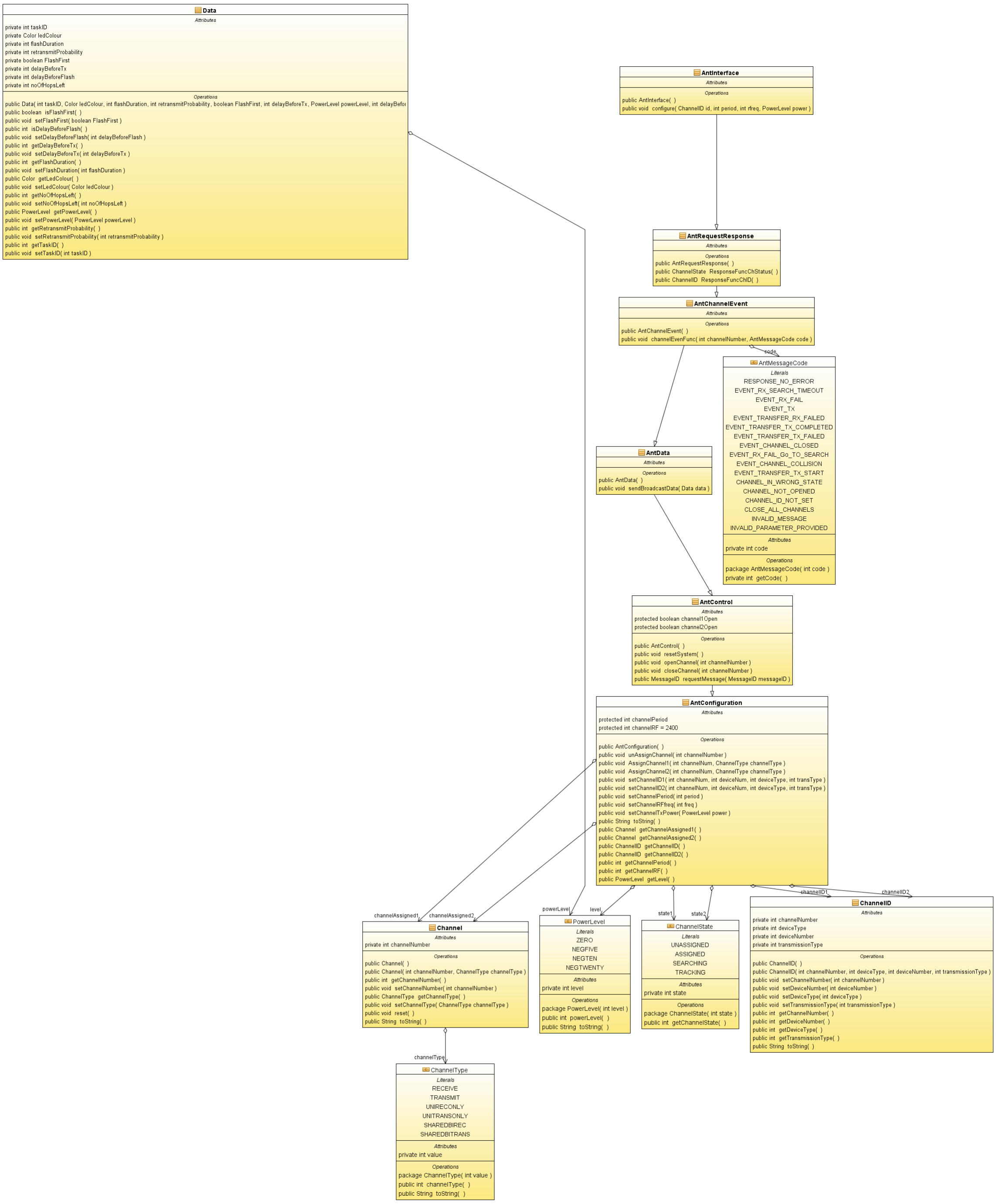


Figure 16: UML class diagram of the ANT Interface

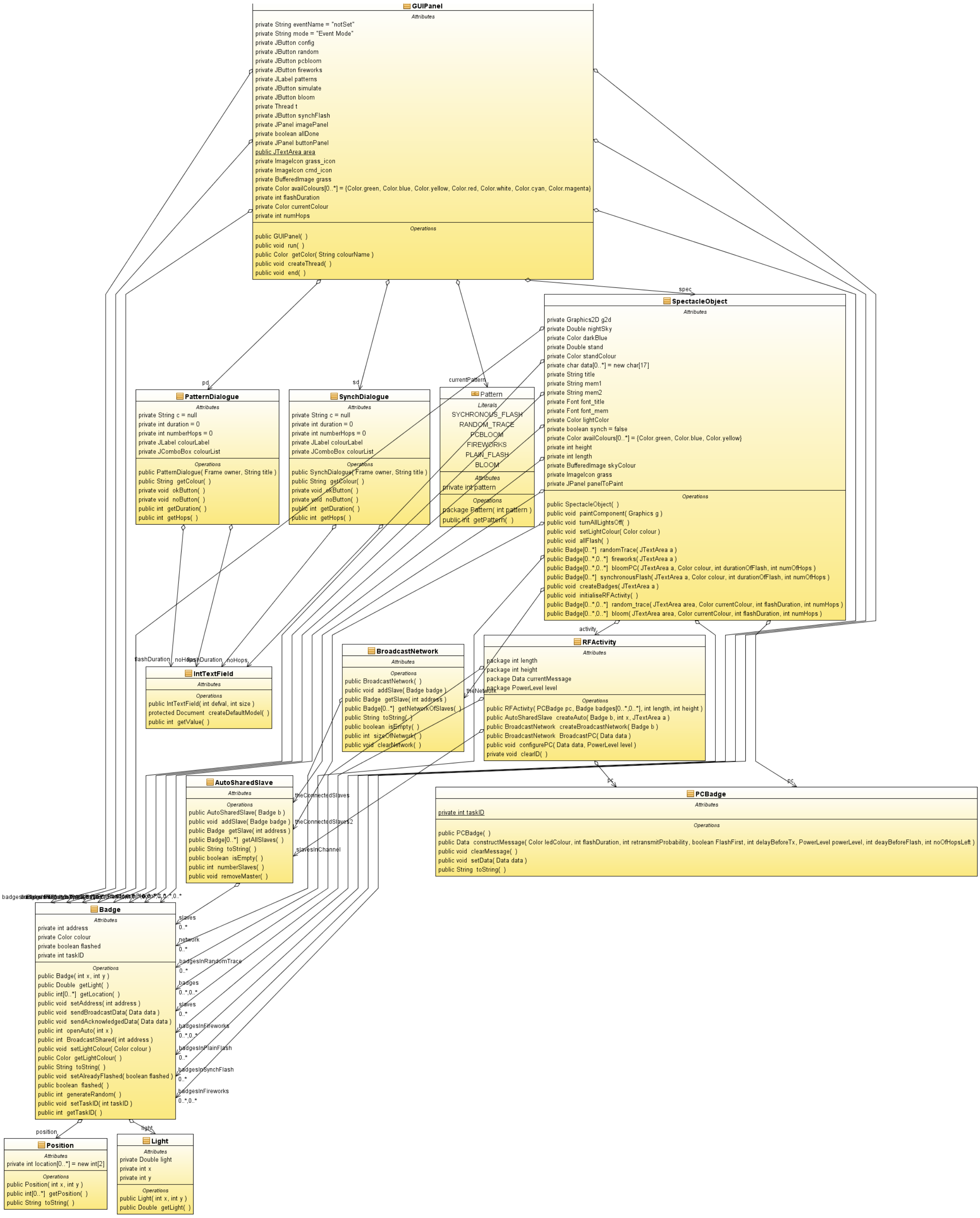


Figure 17: UML class diagram of the Graphics

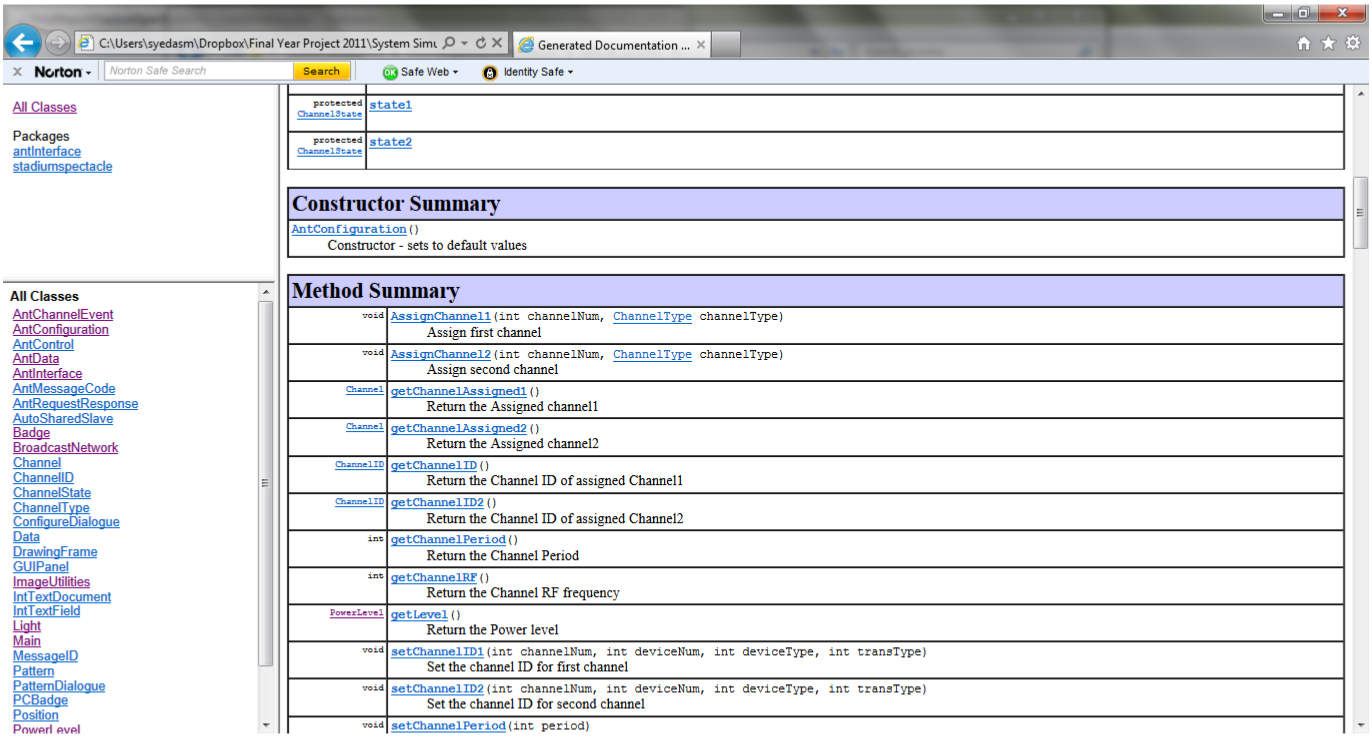
6.5 Challenges Faced

The major challenge that I faced in developing the simulation was, how to pass message around the ANT Protocol, determine which badge is going to flash and when, and connect this to the graphics surface. To overcome this challenge I adopted a "separations of concerns" approach. I kept the Ant Interface and the Graphics surface totally separate and rather than try to pass message and paint the graphics surface at the same time, I decided to pass messages around first and then paint the graphics surface later, after it has been determined which badges should be flashing.

I also found structuring the code in the graphics surface package challenging. The trick to overcoming this challenge was to split the code up into as small classes as possible.

6.6 Javadoc

Javadoc documentation has been generated for the entire software simulation project to help me and any other people who might want to develop the simulation further see which classes are already in the software and the fields and methods that each class has. A snapshop of javadoc is shown below.



7 Results of Simulation

Synchronous Flash

The synchronous flash has just as expected and all of the badges flash simultaneously. The effect has the following look:

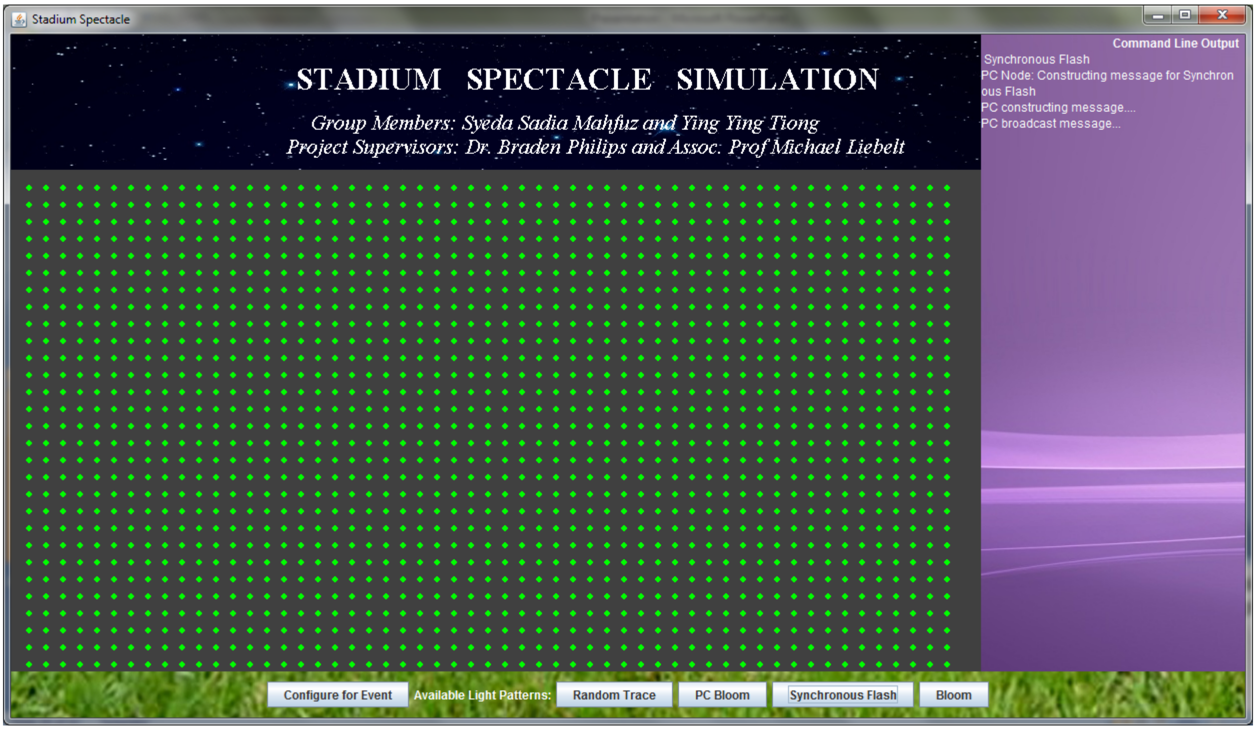


Figure 18: The Synchronous Flash

Bloom

The bloom also works as expected. We observe doughnuts of light that spreads out gracefully form the point of origin, which is the PC node. Figure 19 displays how the bloom effect looks for 4 hops.

Random Trace

The random trace however, yields something a little more interesting. Instead of observing a graceful trace of light, we get something that looks similar to as shown in Figure 20. Instead of getting a random trace, we get more of a random bloom. Light spreads out from the point of origin in a random but circular fashion. One of the challenges in getting the random trace to this stage was to make sure that light that have already flashed one do not do so again. The next challenge would be to make the lights travel in one direction and not outwards

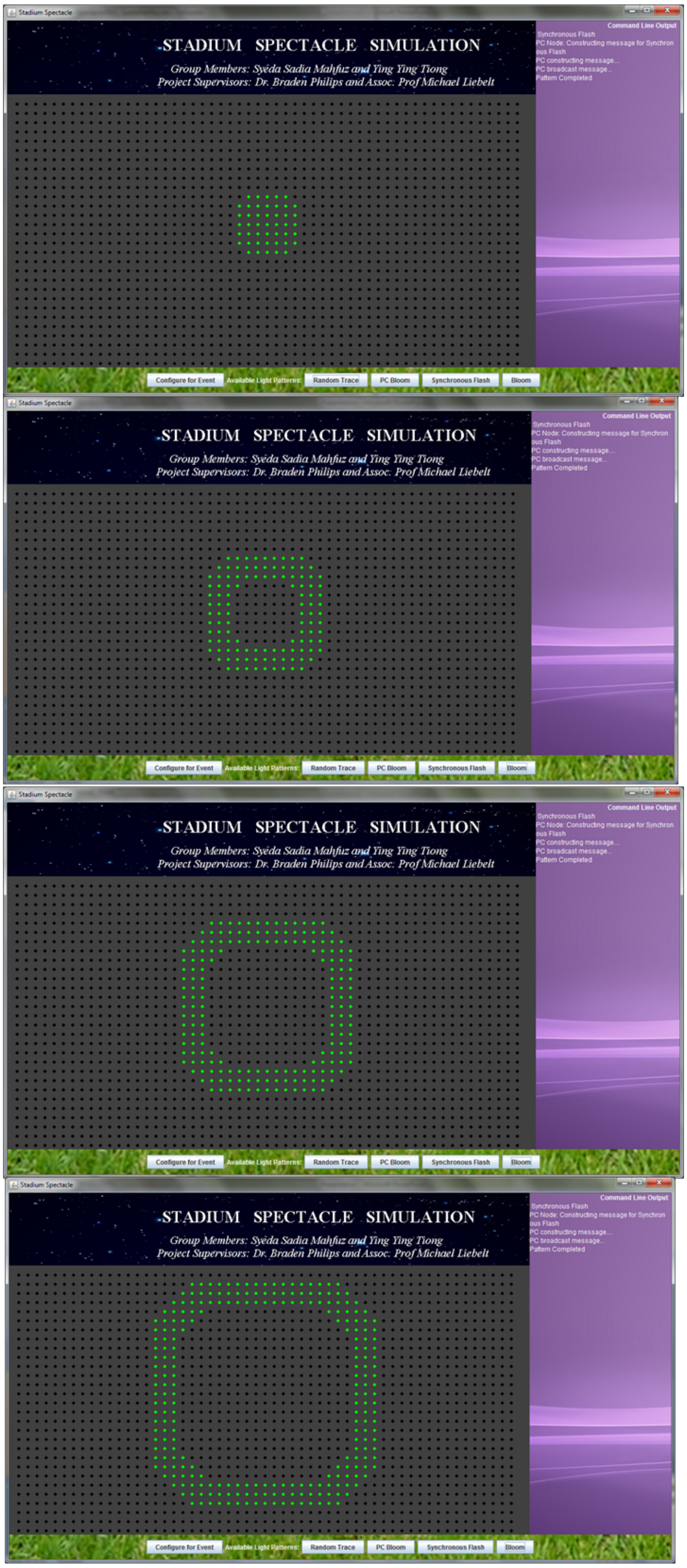


Figure 19: The Bloom

in a circle. Time did not permit me to pursue this challenge this project, however, this is definitely something that future groups might look at achieving.

Figure 20: The Random trace

8 Testing

8.1 Manual Testing

Testing of the simulation software was mainly done via manual testing, through print statements. Whenever a critical section was of code was completed, a driver class was written to test the classes involved. Print statements were included to show me what the class was doing as it entered different methods and sections of the code. For example, when creating a broadcast network, I printed out all of the nodes that joined the network to see whether the nodes I expect to form the network do so.

8.2 Automated Testing

Critical classes were also tested by using JUnit Testing which is supported Netbeans. Basically, these tests created the class being tested and called each method in that test and asserted whether the output of the method matches what is expected. An output from one of the JUnit tests i conducted is shown in Figure 21.

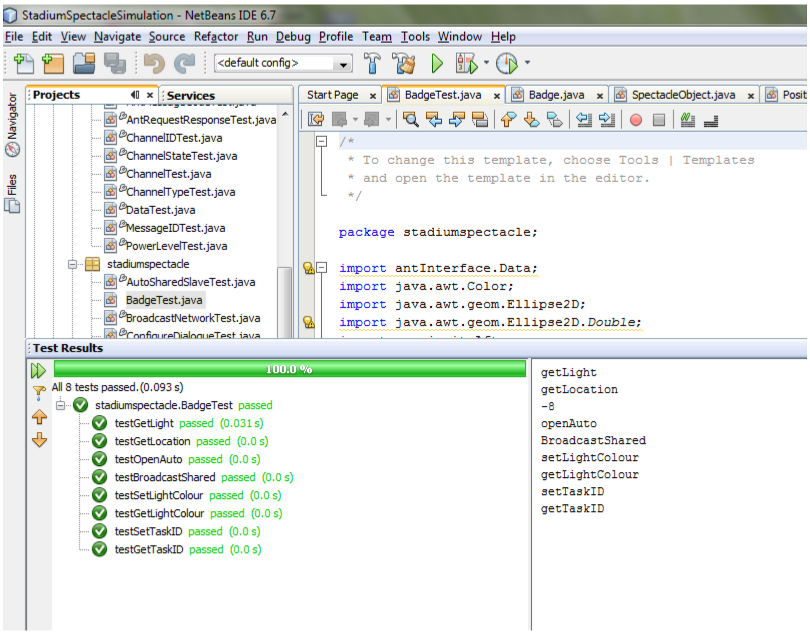


Figure 21: Automated Test Output for Badge Class

Unfortunately, I left Automated Testing until very late in the project. With hindsight, I should have written the JUnit Test cases as I wrote each class and this would have allowed me to detect problems with the code much more quickly and much more easily.

9 Project Management

The project was managed well and the main project goals have been reached. The following sections describe the methods by which the project was managed.

9.1 Allocation of Tasks and Timeline

Since there are two distinct sections to this project, the task allocation was quite simple. Ying Ying was given the responsibility of developing and programming the prototype badges, where as I was responsible for coming up with different designs to form light patterns and developing the simulation software.

The original task allocations for both segments of the project can be found in Appendix A. The original allocation was not very realistic because at that stage, we were still finding out what tasks and resources are required to successfully conduct our project. We were also that we were not competent at allocating complex tasks and in some cases, especially in research and design tasks, we underestimated the time that the tasks would take. As we progressed with our project we developed a better understanding of what needs to be done and we were able to form a more realistic and accurate task allocation. The final task allocation for the System Simulation is shown in Figure 22.

	A	B
1	Task	Due date
2	**Planning Stage**	1st April 2011
3	Choose a wireless network Protocol	
4	Write a simple animated program	
5	Study ANT Network Protocol in detail and identify relevant parts	
6	**Design Stage**	22nd July 2011
7	Design the GUI layout and code the GUI	
8	Devise a network implementation that will allow badges to form ad hoc networks	
9	Determine how the different light patterns will be created	
10	**Software Development Stage**	23rd Septemper 2011
11	Research on graphics development using Java and how to draw multiple graphics objects on one graphics surface	
12	High Level Software Design	
13	UML Diagrams	
14	Model the ANT Network Protocol using Java	
15	Create Badge classes and be able to paint lights onto the graphics surface	
16	Devise the Message Structure with Ying Ying	
17	Implement Synchronous Flash	
18	Implement PC Bloom	
19	Implement Random Trace and Bloom	
20	**Testing and Review Stage**	28th October 2011
21	Junit Testing	
22	Write Javadoc	
23		

Figure 22: Final Task Allocation for System Simulation

9.2 Control and Monitoring

The mechanism used to control and monitor the project and our progress was by way of weekly meetings. At each meeting the achieved progress and upcoming plan would be discussed and the timeline would be updated if required.

Group roles were discussed in these meetings, however, specific group roles were never required to be changed. Task Allocations were changed when the plan was getting behind schedule to ensure that the project goals can still be met, but Ying Ying still undertook the tasks relevant to System Development and I took take of all tasks related to the System Simulation.

One mechanism through which progress was monitored was through interim reports that were produced at regular intervals. These reports were in addition to the Stage 1 and Stage 2 Progress Reports. A list of interim reports produced is given below:

- **Wireless Network Report** - an evaluation of available wireless network protocols and discussion of which is most suitable for this project
- **Network Set up** - discussion of types of light patterns we want to create and how the wireless network could be set up to achieve them
- **Dialogue Between Node and Commands to Write** - a high level discussion of how the ANT nodes can communicate and what sorts of messages would need to be written for each light pattern
- **Transmission Range Experiment** - The results from the transmission power versus transmission range experiment
- **Short Report for Current Prototype** - discussion of the message structure and how the current prototype badges are set up

9.3 Risk Management

The Original Risk Assessment can be found in Appendix B.

Upon reflection the original risk assessment represented a pretty fair assessment, and most of the identified risks were encountered to some degree. However, the risks that had the most effect on our project were and the strategy we used to deal with these risks is discussed below:

1. **Hardware components are not delivered in time**
   The arrival of the ANT development kit was delayed by one month. During this time, LED driver circuits were designed and experiments were carried out by Ying, so that by the time the ANT development kit arrived, she would be ready to work on creating the badges straight away.

2. **Unforeseen Technical Difficulties**
   A lot of the tasks that were undertaken during the project were totally new to us and at times we were stuck on how to proceed further. There were times when we would form one opinion and design, only to find out that our ideas do not work as expected. Fortunately, during these times, our supervisors were willing to share their experiences and give us hints to guide us towards the right track. We also created an account in ANT forums, to be able to contact other developers who use ANT. Also, a lot of support was found from workshop staff when Ying Ying was constructing the prototype badges.

3. **Destruction the component**
   One of ANT radio modules were accidentally destroyed when constructing the prototype badges. Unfortunately, these things are often inevitable in hardware work, and in our case it was even more of a risk since the ANT radio modules are so small.

4. **Project falls behind schedule**
   Due to our inexperience in scheduling tasks in a big project, we found early on the our project was falling behind schedule. Some tasks, especially design tasks took much longer than anticipated. As soon as we realised this we revised our timeline and took out tasks that were of less importance. For example, in the simulation, there was a plan originally to add in real behaviour of badges, such that sometimes transmitted messages will not get received. However, this task was taken out since more time was required in researching the ANT Protocol design of light patterns.

9.4 Budget

Each group member was allocated a budget of $250, meaning the total available budget for this project was $500. Figure 23 displays how this budget was used.

Final Year Student Group Budget (2011) - School of EEE

Project Name: Stadium Spectacle

Student 1: Ying Tiong
Student 2: Syeda Mahfuz
Student 3:
Student 4:
Student 5:

Supervisor: Braden Phillips/Mike Liebelt

Student Budget: $250.00
Project Code:
Location:

Date:	Part #:	Description:	Qty:	Unit Price:	Total Price:
	COM-09853 (SparkFun)	Triple Output LED RGB - Clear - 25pcs	1	$19.95	$19.95
	DEV-09266 (SparkFun)	LilyPad Arduino 328 Main Board	3	$21.95	$65.85
	DEV-09716 (SparkFun)	FTDI Basic Breakout - 5V	1	$14.95	$14.95
		Store Parts	1	$30.00	$30.00
					$0.00
					$0.00
					$0.00
				Total	$130.75
				Remaining	$369.25

Figure 23: Budget for our Project

In addition to the items mentioned above, the items listed in Table 2 were also purchased. These items however were sponsored by our supervisor, and the School of Electrical and Electronic Engineering.

Table 2: Items purchased by sponsorship

Item	**Sponsor**	**Cost**
Ant Development Kit	Dr. Braden Philips	$700
Arduino Uno Board	School of EEE	$29.95
Eleven (100% Arduino Uno Compatible)	School of EEE	$39.95
Total		$769.90

9.5 Configuration Management

The resources used for configuration management of the project are:

- Dropbox
- SVN
- Logbooks

Dropbox was by far the most useful tool of the three resources. The Dropbox is an extremely user friendly tool. We both installed the Dropbox on our computers and put our files in it and the Dropbox syncs files automatically as they are changed. Ying Ying and I were able to share files very effectively and easily. We both put our files for the project in the Dropbox and we were able to look at each others work as it was being updated.

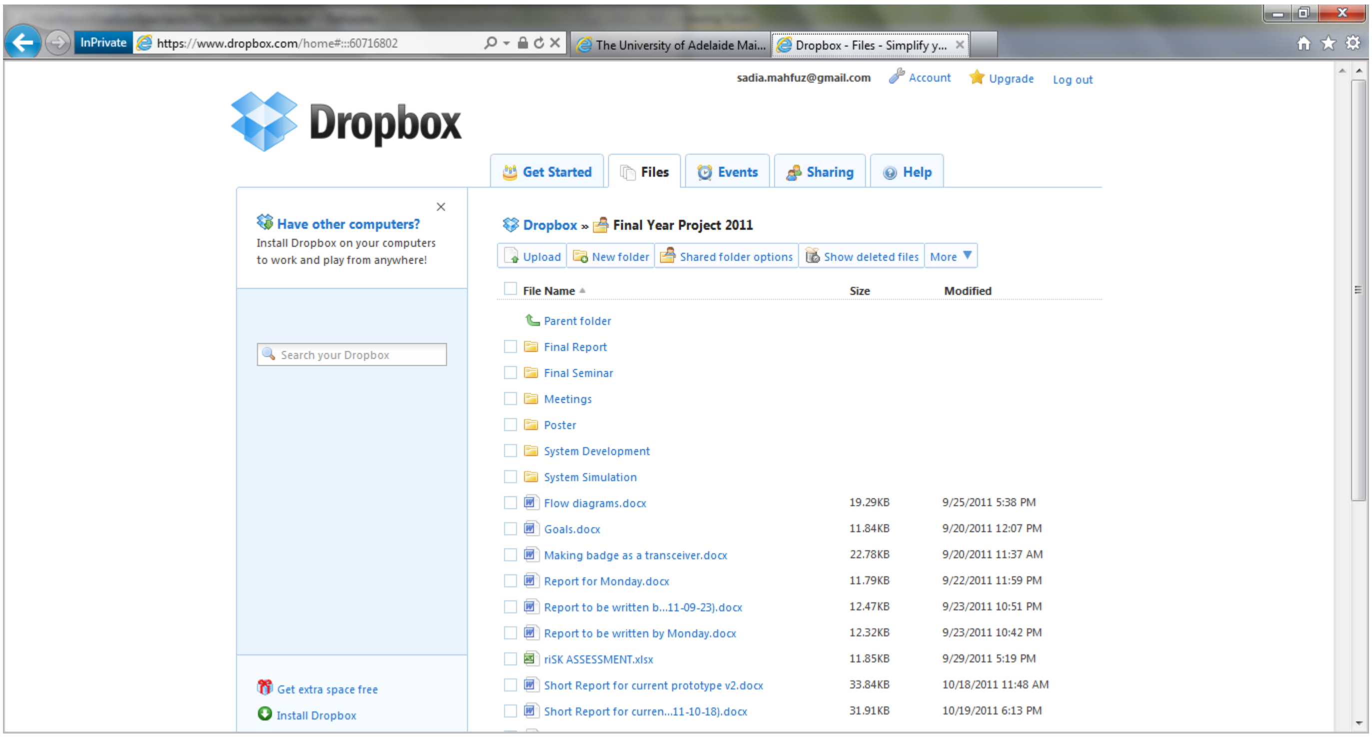


Figure 24: The Dropbox File Structure

10 Final Cost of System

This section attempts to estimate the cost of our system for an event that required 100 000 badges.

PC Base Station

The PC base station requires:

- ANT USB Stick: Cost = $27.96 if hundreds are bought (from SparkFun)
- A PC: Cost = $500 (for a small notebook)

Total Cost of PC Base Station = $527.96
This figure is well below the maximum value of $10000 as required in the specifications.

The Badge Nodes

The PC base station requires:

- ANT Radio Module: Cost = $3.56 (from SemiconductorStore)
- AtMega MCU: Cost = $2.50
- Other Components (LEDs, resistors) = $2.00

Total Cost of each Badge Node = $8.06

11 Reflection on Project

11.1 Strengths and Weaknesses

The strengths of the project are:

1. **The system is simple yet spectacular**
   The system is built upon quite a simple design, yet the light effects that can be created are quite spectacular.

2. **System is small and ultra low power**
   The ANT nodes are really small and they consume very low power. One of the main reason for choosing ANT in this project were these qualities of ANT, which are an asset to have.

3. **System is wireless**
   There is no mess with wires and this makes the system look much neater

4. **Employs low cost ad hoc networking**
   The system uses low cost ad hoc networking. Ad hoc networking means that there is no need for central access point in the network, and thus, there are no single points of failure within the system. This makes the system more reliable and flexible.

The weaknesses however, are:

1. **System has to rely on at least on platform the PC node**
   As has been discussed throughout this report, the PC node constructs the message and it initiates the passing of messages for any given light pattern. However, having multiple PC nodes within the stadium in an event, will somewhat compensate for this weakness, because of one PC node breaks down, there will be others that can still keep working.

2. **No market research conducted**
   A major weakness is that we have not conducted any market research into this type of system. We do not know whether people would be interested in this sort of system for their event.

3. **How will it really look**
   Another major weakness is that we do not really know whether the final system would look good. We have couple of prototype badges to give us an idea of how each individual badge might look. We also have a simulation that gives us an idea of how the system might look on a bigger scale. However, until we create a hundred thousand badges, put them in a stadium and run our light patterns, we cannot tell whether the system would really look spectacular.

11.2 Other Applications

We have used ad hoc networking in our project to create a fun system, however, there are many applications that can be constructed using similar principles. An example of such a system is tracking tools. The badges can be created such that a badge is given to a child and to the child's parent, and the parent can keep track of the child. Another example is unmanned robots. These badges can be attached to thousands of tiny robots. The small robots can communicate to each other through the badges and together they can build up a huge robot.

11.3 Social and Cultural Impact

The other applications leads us to think of the social impact that ad hoc networking might have on the community. There are many ways in which society can benefit from such applications. For example, small badges can be created to help parents keep track children in a crowded area such as the Royal Adelaide Show or shopping centres. However, it can also have a negative impact on society. People with bad intentions could hack into the system, and try to stalk people.

11.4 Environmental Considerations

The badges that have been developed are disposable products and hence, environmental considerations need to be made. The following steps can be taken in future to reduce negative impact on the environment that the badges can have:

1. Make the badges even smaller and reduce the number of component by as much as possible
2. Use lead free solder
3. Use environmentally friendly and rechargeable batteries

12 Conclusion

12.1 Summary of Project Outcomes

The project has produced the following deliverables:

- 2 prototype badges
- 1 PC base station
- A PC simulation with hundreds of nodes
- An estimated final cost of the system

The Simulation has been successful because it can create the three light patterns that were originally outlined in the requirements with hundreds of nodes. The Simulation has been a useful tool since it has complemented the hardware development by trying out different message structures and helped decide what sorts of parameters are useful in the message structure.

Although lack of time has prevent more that 2 prototype badges being constructed (the original requirements specified four badges) the System Development has also been successful. The two badges can successfully operate in the souvenir mode. The can also connect to the PC node and form the synchronous flash light effect.

The estimated final cost of the system also falls within the range of values that were required in the original requirements.

12.2 Lessons Learnt

Undertaking this project has given me a valuable insight into wireless networking and in particular ad hoc wireless networking, which was a completely new concept to me. The project has also helped me to develop software modelling and structuring skills.

12.3 Summary of Future Work

In the short term the project can be improved in the following ways:

- Design a way to calibrate the badges so that they have a crude sense of location. This will allow them to pass create light effects that travel in a certain direction only.
- Construct more prototype badges and implement the bloom and random trace effects in them

- Develop more light effects to be implemented

In the long term the following can be considered:

- Add sensors to the badges so that they can detect and flash to a particular sound, e.g. drum beats
- Integrate badges with GPS so that badges know their location which will allow them to form more complex light effects and even images!
- Make the badges even more power efficient and increase the battery life as much as possible
- Develop a secure network in which the badges operate

Appendices

A Original Timeline

Table 3: Detailed Work Breakdown Structure - Semester 1

	Task Description	Duration	Due Date	Person
1	Choose Wireless Network Protocol	15 hours	21/03/2011	Both
2	Write a simple animated programmed showing to lights flashing to practise graphics and animation skills	10 hours	28/03/2011	Syeda
3	Choose and order relevant hardware components	5 hours	01/04/2011	Both
4	Estimate a top level power budget(start at AAA battery and work downwards)	5 hours	04/04/2011	Ying
5	Study the network topology in detail and determine which topologies will be implemented and which commands will be used	20 hours	11/04/2011	Both
6	Design the circuit configuration for prototype badges. The maximum distance the radio module inside the badges can send signals is 1 m and it must be able to drive 3 leds.	10 hours	18/04/2011	Ying
7	Complete a system design for the software simulation. Consider how deeply the network protocol needs to be coded. This depth should correspond to the level of detail that is used in the hardware implementation of the system.	12 hours	20/04/2011	Syeda
8	Design the look and layout of the badges, consider how many leds are to be on a given badge	3 hours	22/04/2011	Ying
9	Implement the badges graphically in the simulation to look as close to their harware counterparts as possible	5 hours	25/04/2011	Syeda
10	Make a design for the software modules that will reside in the badges	7 hours	28/04/2011	Ying
11	Design a GUI mock - up for the simulation	3 hours	28/04/2011	Syeda
12	Write the LED Driver Circuit	7 hours	09/05/2011	Ying
13	Construct the hardware badges	7 hours	16/05/2011	Ying
14	Create a software model of the network protocol	25 hours	30/05/2011	Syeda
15	Program microcontroller to get one LED flashing	12 hours	30/05/2011	Ying

B Original Risk Assessment

Risk Level Estimating

The potential risks for this project are listed in Table 1. The listed risks are rated with its risk level based on the risk level estimator (based on the tables in Appendix B which is taken

Table 4: Detailed Work Breakdown Structure - Semester 2

	Task Description	**Duration**	**Due Date**	**Person**
16	Wirelessly connect the PC to the badges such that PC can send messages to the badges and badges respond as asked	18 hours	18/07/2011	Ying
17	Implement the badges and PC base station such that they can pass messages to each other as per the network protocol	24 hours	25/07/2011	Syeda
18	Implement different system configurations (e.g. let each network topology be a different configuration) as well as different light patterns for each configuration	24 hours	20/08/2011	Syeda
19	Wirelessly connect two microcontrollers and get messages passing between them	25 hours	22/08/2011	Ying
19	Get the connection between PC and badge to badge happening smoothly to create various light patterns	12 hours	12/09/2011	Ying
20	Add in hundreds of badges to the simulation	18 hours	19/09/2011	Syeda
21	Code debugging and testing: Think about any situation that can break down the system and try to cope with those situations.	15 hours	26/09/2011	Ying
22**	Add in more badges and light patterns to the system	15 hours	17/10/2011	Ying
23	Estimating cost for applying the system into real situation	1 hour	17/10/2011	Ying
24	Final testing and Review for Simulation	12 hours	17/10/2011	Syeda
25	Project Exhibition	4 hours	28/10/2011	Both

from [9]).

Note that ** indicates OH & S Risks

Table 5: Risk Level Estimating

Risk	**Likelihood of harm**	**Severity level**	**Risk level**
Group members are side-tracked from work	very likely	moderate	very high
Project falls behind the schedule	very likely	extreme	very high
Hardware ordered do not match with the requirement	very likely	extreme	very high
Overbudget	very likely	moderate	very high
Loss of motivation due to getting bored or finding the task too difficult	likely	moderate	high
Hardware components are not delivered in time	unlikely	moderate	medium
Software larger than anticipated	likely	slight	low
Unforeseen technical difficulties	likely	slight	low
**Incorrect working posture that causes shoulder and back pain	very likely	slight	low
**Safety during soldering	likely	slight	low

Risk Management

1. **Group members are sidetracked from work:** Set a due date for each task. Group members are urged to finish every task before the due date. Avoid scheduling major tasks during known exam and test times. Be prepared to put in the few extra hours should there be a sudden surge in workload later on in the semester

2. **Project fall behind the schedule:** Replan the timeline by reducing some unnecessary task. Perform tasks in parallel if possible.

3. **Hardware ordered do not match with the requirements:** Do research from the datasheet or consult the manufacturer.

4. **Over budget:** Try to look for cheaper solution.

5. **Loss motivation due to getting bored or finding the task too difficult:** Group members will encourage each other stay focused on the project. Members will also help each other out should task prove to be too difficult for just one person.

6. **Hardware components are not delivered in time:** Allow enough time in schedule for reasonable delays.

7. **Software larger than anticipated:** Implement the most necessary functions first. Initially purchase a microcontroller such that is has more programmable memory than we think is required.

8. **Unforeseen technical difficulties:** In timeline, schedule in some weeks that are relatively free to allow for such delays

9. ****Incorrect working posture that causes shoulder and back pain:** Always aware on the working posture during the week

10. ****Safety during soldering:** Never leave the hot soldering iron unattended. Wear safety glasses while soldering.

References

[1] "ANT - the power of less," March 2011; http://www.thisisant.com.

[2] C I Diamond"How ZigBee Works" Nov 2005; http://homepage.uab.edu.

[3] "FindChips.com" March 2011; http://www.findchips.com.

[4] "Formerly Farnell Element14" Nov 2005; http://au.element14.com.

[5] "Nordic Semiconductor" Nov 2005; http://www.nordicsemi.com.

[6] "Semiconductorstore.com - A Division of symmetry Electronics Corp. " March 2011; http://www.semiconductorstore.com.

[7] "Texas Semiconductor" March 2011; http://www.ti.com.

[8] "ZigBee Alliance" March 2011; http://www.zigbee.org.

[9] http://www.scribd.com/doc/50463646/14/Table-E-3-A-simple-risk-estimator

[10] "Menus, Layouts, Styled Text, and Simple Geometry" http://java.sun.com/developer/technicalArticles/GUI/swingmenus/

[11] "Java 2D" http://download.oracle.com/javase/tutorial/2d/index.html

[12] "Stopping a Thread" http://www.exampledepot.com/egs/java.lang/StopThread.html

[13] ANT Message Protocol and Usage," January 2011;http://www.thisisant.com/images/Resources/PDF/1204662412_ant%20message%20protocol%20and%20usage.pdf

[14] 'Ad Hoc Networks Research," April 2011; http://www.csse.monash.edu.au/~carlo/adhoc.html

[15] http://www.sparkfun.com/

[16] "Java examples (example source code)" 2009; http://www.java2s.com/Code/Java/CatalogJava.htm