Investigating an Integrated Inertial Gesture Recognition System and Vibrotactile Display

A thesis submitted to the
University of Dublin, Trinity College,
in fulfilment of the requirements for the degree of
M.Sc. in Computer Science.

Joseph Mc Knight

Ubiquitous Computing,
Department of Computer Science,
Trinity College, University of Dublin

May 2006

DECLARATION

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work.

____

Name,
Date

PERMISSION TO LEND AND/OR COPY

I, the undersigned, agree that the Trinity College Library may lend and/or copy this thesis upon request.

____

Name,
Date

ACKNOWLEDGEMENTS

To Gavin Doherty, my project supervisor for his guidance and advice throughout the last year. Thanks for always making yourself available for project meetings to discuss the direction of the project and help with any issues.

To all Ubicom classmates, two great years working with all you.

To Cormac Driver, thanks for your help in getting any equipment needed for the project without fuss or delay.

To James Garland, for his advice at the start of the year on what was required to build the tactile array.

To the lab technicians, thanks for always being accommodating when I needed to use your equipment.

Abstract

As the ubiquitous computing paradigm that is predicted for the future is brought closer by technological advances, designers of new smart homes, mediated spaces and sentient computer systems will have to consider new techniques to interact with users. By moving away from the traditional GUI and looking at new interactions techniques that might be more suitable for these future environments, we can make this transition more manageable and perhaps natural. The work in this dissertation will take a look at an integrated gesture and tactile feedback system to access its suitability as an interface to future systems.

To test out a new interaction I have developed a gesture recognition system to function as an input mechanism for users. Gesture was chosen due to its advantages over GUI's and voice in certain circumstances. The gesture recognition is based on processing Euler angle data that is fed to the system using an XSens MT9 motion sensor. In total 8 distinct gestures are recognisable by the system which can be used as an input interaction mechanism.

The usefulness of simple tactile feedback, as provided by phones etc, is widely accepted. This is because of its discreteness in certain environments where noise is unacceptable. By developing a tactile array I hope to show that more meaningful interactions can be facilitated and developed by having the ability to convey information through tactons which can contain a lot more information than the simple vibration function of a phone.

To test the viability of the system a number of user tests where performed with the integrated system and the individual components. The results of which suggest that the integrated system has merits as an interface mechanism that could run in tandem with other interaction techniques or used in isolation.

Table of Contents

Chapter 1: Introduction……………………………………………………….1

1.1 Introduction………………………………………………………...1

1.2 Project Goals……………………………………………………….2

1.3 Organisation of Dissertation……………………………………….3

Chapter 2: Prior Art…………………………………………………………..4

2.1 Inertial Gesture Recognition……………………………………….4

2.2 Gesture in Multimodal Interaction……………………………….....6

2.3 Tactile Perception…………………………………….....………10

2.4 Investigating Tactile Feedback……………………………...……10

2.5 Wearable Vibrotactile Displays………………………….……….13

Chapter 3: System Hardware Components…………………………………15

3.1 High Level Component Design……………………………….....15

3.2 MT9 Overview……………………………………………………16

3.2.1 MT9 Internal Sensors……………………………………...17

3.3 IPAQ 5500…………………………………………………….....18

3.4 Total Remote IR Dongle…………………………………………18

3.5 Keil MCBx51 Development Board……………………………...19

3.6 ULN 2003an Darlington Array……………………………..... 20

3.7 SAM A100 Vibration Unit………………………………………..21

Chapter 4: System Analysis and Design……………………………………22

4.1 Overall System Analysis & Design…………………………….....22

4.1.1 Brief System Description…………………………………23

4.2 Design of Gesture Subsystem……………………………………24

4.2.1 Explanation of UML Class Diagram……………………...26

4.3 Design of Tactile Array…………………………………………...27

4.4 Design of Collect Data Subsystem……………………………….28

Chapter 5: Gesture Recognition…….………………………………………29
5.1 Definition of a Gesture……………………………………………29
5.2 Sample Data Stream………………………………………………30
5.3 Pre Gesture Recognition Activity Detection……………………...31
5.3.1 Pre Activity Detection…………………………………….32
5.3.2 Activity Detection.……………..…………………………...33
5.4 Gesture Tracking……………………………...…………………..34
5.5 Gesture Recognition……………………………….…………….35

Chapter 6: Building the Vibrotactile Array………………………………...…37
6.1 Circuit Design………………………………………………….37
6.2 Serial Communication……………………………………………39
6.2.1 Implementing Rs232 Communication…………………..….40
6.2.2 Setting the Serial Port Mode……………………………...40
6.2.3 Receiving a character from the Serial Port………………….41
6.3 Generating Display Patterns……………………………………...41

Chapter 7: InfraRed Signal Generation……………………………………….44
7.1 Generating an IR Signal…………………………………………..44

Chapter 8: Testing…………………………………………………..………..46
8.1 Testing the Vibrotactile array……………………………..…...46
8.2 Description of User Tests…………………………….…..……46
8.3 Testing Feedback System for Remote Control……………..……49
8.3.1 Analysis of Testing on Array Design 1……………...……49
8.4 Conclusion of Initial Testing……..……………………….…..……51
8.5 Testing of Redesigned Tactile Array……………………….…....51
8.6 Conclusion of Testing………………………………….…….…..53
8.7 Gesture Testing……………………………………………..…….55
8.8 System Testing………………………………………….....……55
8.8.1 Test Format………………………………………………57
8.8.2 Game Restrictions……………………………………...57
8.9 Analysis of System Testing...57

Chapter 9: Evaluation………………………………………………………..58
9.1 Gesture Recognition……………………………………………58
9.2 Vibrotactile Array………………………………………………...61

9.3 Integrated Gesture and Tactile Display……………………………..62

Chapter 10: Conclusions and Future Work…………………………………….63

10.1 Summary…………………………………………………………….63

10.2 Future Work…………………………………………………………64

LIST OF FIGURES

Figure 3.1 – Hardware Component Design……………………..……………………16
Figure 3.2 MT9 Sensor……………………………………………………………16
Figure 3.3 Griffin Total Remote IR Dongle……………….…..…………………18
Figure 3.4 MCBx51 Development Boar....……………………….………….…….19
Figure 3.5 ULN2003an…………………………………………………….…….21
Figure 3.6 Vibration Unit………………………………………………….…….21
Figure 4.1 Layered Architecture of Integrated System……………………….…....23
Figure 4.2 Recognise Gesture Subsystem…………………………………….……24
Figure 4.3 UML Class Diagram of Gesture Subsystem…….…………….……..25
Figure 4.4 Embedded CPU cycle executions………………….……………….…..27
Figure 4.5 Motion Tracker Object……………………….……………………28
Figure 5.1 Example Data Streams…………………….…………………………31
Figure 5.2 – Determine Start of Gesture Function…………………………………32
Figure 5.3 Detect Movement Function……………………………………………..33
Figure 5.4 Recording Gesture End Function Excerpt………………………………35
Figure 5.5 Gesture Comparison Function Excerpt…………………………………36
Figure 6.1 Circuit Diagram…………………………………………………………37
Figure 6.2 Darlington Arrays……………………………………………………….39
Figure 6.3 Setting the Serial Port Baud Rate……………...……………………….41
Figure 6.4 Left to right display pattern……………………………………………..42
Figure 6.5 Hardware Delay Function………………………………….……….43
Figure 7.1 IR Signal…………………………………....………………….……..45
Figure 8.1 Test Results…………………………………...………………..54

Chapter 1: Introduction

As Mark Weiser’s vision of ubiquitous computing [1] becomes more of a reality and computing trends continue towards distraction free pervasive computing [2], the need for new interaction techniques also arises. Weiser’s vision of computer systems beginning to disappear into the background to the extent that people become unaware of there existence, due to sentient computer systems [3] operating on their behalf, or in the future when possibly a new plethora of pervasive computing system are competing for our attention, new interaction techniques for users will also need to be defined.

Examples of current pervasive/ubiquitous computing projects like [2] [4], where pervasive human centred computing is the goal, rely on a sentient computer system that interacts with the user while they occupy a mediated space [5]. In [4] the interaction with Oxygen is through spoken language and visual clues instead of a keyboard and mouse to make human-machine interaction more natural, but fails to take into account that a user might require a more private interaction as stated by [3]. If we move away from keyboards and mice and personal displays in these mediated spaces in favour of a more natural interaction method with sentient computing systems, then we loose all privacy unless we are the only persons occupying the space.

The interaction may become more natural from a communication perspective but a lot of the time humans partake in private interactions where the communication is on a 1 to 1 basis. As sentient computing systems become more “human-like” and our interactions with them become more human through natural language, then I would propose that we would like to ensure our styles of interactions with these systems can be tailored to our needs.

Project Goals

Investigating a new interaction technique was the basis of my research topic. To enable me to carry out some research in this area I have designed and built two separate but co-operating systems to investigate these different types of interaction. The first part was a gesture recognition system that could be used as the user's main interaction mechanism to control electronic equipment that is based on IR. This is of course for demonstration purposes only and could be used as a control/interaction mechanism to other applicable systems such as the sentient systems just discussed.

For the final part I built a tactile array to use as the feedback mechanism to the user. The principal with the tactile array was that feedback could be given to the user by utilising their cutaenous modality. This achieves two objectives, the first is a means of conveying information to the user through the use of a tactile display and the second is that this information can be conveyed in a private manner to the user while not impinging on other modalities. This type of interaction could be used in mediated spaces in conveying information to users in a private manner if warranted or purely as a new interaction technique when other modalities are overloaded.

Organisation of Dissertation

As stated earlier the work of my dissertation is to research new interaction techniques from use in human-machine and machine-human interactions. I begin in chapter 2 with a selected literature review of the current state of the art in gesture recognition and tactile displays. The purpose of this chapter is to give the reader an understanding of the issues and work that is being carried out in these areas that I researched during the design phase of both systems.

Chapter 3 gives a description of the hardware components that where used to build the gesture recognition and tactile display systems. This chapter will give the reader an understanding of what was involved from a hardware perspective to build both systems.

In chapter 4 I show an example of the data stream that the gesture recognition is performed on. The purpose of this is to give the reader an example of the data stream to the reader to make their understanding of the gesture recognition algorithm clearer while reading further chapters. I then explain how both systems where designed and go through the process from a description of the gesture recognition system through to a UML class diagram for the system. This chapter also looks at the software design of the other major components that had less coding but was more hardware based, such as the tactile array.

Chapter 5 goes through the gesture recognition algorithm and explains how the gesture recognition is performed by the system. It is explained from the initial data capture from the data stream, to the activity detection algorithm that monitors the data stream looking for the possible start of a gesture, and finally the recognition process. Chapters 6 and 7 goes through the design and build of the tactile array which is mainly a discussion of hardware issues involved in building the array and also how the IR (Infra-Red) component was also built.

Chapter 8 goes through the testing that was carried out on the tactile array as an individual component to ascertain the best temporal parameters to be used in displaying patterns with the array. The testing then finishes with a discussion of an overall system test and analysis of this testing in relation to the project goals described.

Chapter 9 is an evaluation of the individual components of the project and the overall system. I go on to discuss the goals of the project and evaluate my achievements against them.

Chapter 10 is a conclusion of my dissertation with a discussion on the possible applications and uses of the system as a whole and also the individual components. The chapter finishes with a discussion of possible future work.

Chapter 2: Prior Art

I could not uncover any academic research investigating gesture recognition and vibrotactile arrays as parts of an overall integrated system, so my prior art will be divided into sections for each topic area. Most of the academic literature on gesture recognition is divided into two distinct areas, one is based on using computer vision by processing captured video sequences and the other is based on inertial sensing of gestures based around accelerometers and gyroscopes. The research in the area of tactile displays is centred on the use of the cutaneous sense to convey information to the wearer. These displays usually vary by the target area of the body the display is worn and on and the actual dimensions of the display.

2.1 Inertial Gesture Recognition

Benbasat's [6] Master's dissertation looks at developing an Inertial Measurement Unit (IMU) for User Interfaces. Benbasat recognises the advantages that inertial sensing devices provide over the traditional vision based approaches in terms of size and robustness plus the fact that they can be made wireless. He claims that the use of low cost inertial sensors in applications can be a more flexible sensing modality. For today's application developers no framework exists that allows them to define the set of gestures they would like to use with their application without the need to develop a mechanism for gesture recognition for each application. His vision is for a designer to quickly specify gestures for detection and the desired action to be taken and not have to worry about developing a gesture recognition system.

His proposed solution was to develop a six degree of freedom IMU based on three single axis gyroscopes and two two axis accelerometers to be used for gesture recognition. The low cost of building this device is a key requirement of keeping project and another is low complexity for recognition of gestures due to the processing requirements that would be placed on any device that used the system. To this extent Benbasat ruled out using any of the more high cost techniques such as using a Kalman Filter [7] and Hidden Markoff Models HMM [8]. Instead he uses his own activity detection and recognition algorithms to parameterise gestures to the smallest atomic gesture that cannot be decomposed further and avoids the processor intensive methods mentioned. He claims that a designer can compose his own gestures, which are composed of the smaller atomic gestures to define the gestures of interest that should occur.

Benbasat showed using a sample application that gestures could be derived from many atomic gestures, although changes where required as some gestures where the sum of the same atomic gestures causing misrecognition. He also showed by comparing a HMM with the same data stream as his algorithm that the HMM could not accept data at even half the update rate of his algorithm. This showed that accurate gesture recognition can be done using propriety techniques with lower processor overhead than traditional but more generalise able techniques.

Urban, Bajcsy, Kooper [9] investigated the use of Orientation Sensors for the recognition of hand arm gestures. Orientation sensors are the same as the IMU in Benbasat's thesis but with on board processing to give Euler angles from accelerometer and gyroscopes. The system was designed to be used for the control of unmanned aerial vehicles (UAV) on an aircraft deck. The reason why inertial gesture recognition was consider over other methods such as speech or video, was due to the unfavourable conditions of noise and clutter of other vehicles and people on deck eliminating these techniques. They also investigate the issue of real time recognition of gestures using multiple orientation sensors attached to the arm. They wanted to show that the selected gestures where repeatable from a recognition sense. This is necessary to show that the defined gestures produce the "same" data stream each time it is performed, therefore allowing repeated recognition.

The authors propose that by carefully placing three orientation sensors on the arm should allow them to recognise the twenty different gestures that are necessary for the control of the UAV's. They use a dynamic time warping (DTW) metric method to test the repeatability of the gestures by comparing the data stream axis by axis and then the complete gesture. Similar metrics would indicate that users are able to perform the same gestures and any gestures whose metric indicate similar reading could possibly be misclassified.

There evaluation shows that by varying the position of the sensors on the arm helped to distinguish the classification of the gestures. They also determined that 2 sensors per arm, one on lower one on the upper, was enough to differentiate between the twenty possible gestures needed. They show that the placement of each sensor along the arm is critical to the content, accuracy and robustness of the gesture recognition. To test the repeatability of gestures the DTW algorithm calculates the difference between two gestures by comparing the data angle by angle for each axis and gives a difference metric at the end. The tests showed that the gestures where repeatable and conclude that the gestures could be recognised using a template based method.

The authors main contributions was to show that gesture recognition could be performed using orientation sensors and that they could be classified by matching recorded gestures with a template for that gesture.

2.2 Gesture in Multimodal Interaction

In [10] Brewster et al investigated the use of gesture recognition as the interaction technique for selection of items from a radial pie menu while on the move. They state that characteristics of today’s mobile computers such as small screen size and limited input, typically with a pen, makes interacting with such devices difficult while on the move. When users are walking most of their attention is spent navigating the physical environment so the attention that can be dedicated to interactions with a portable computer is limited. A multimodal “Eyes-Free” technique would be the preferred input technique in these situations where other modalities cannot be used. Computer vision is not an option with mobile devices and even if speech recognition rates in noisy environments could be improved the computation needed would be a significant drain on a mobile device.

The proposed solution was to use head gestures as the means of choosing an item from a pie menu for control of an application. The user selected items in the pie menu by nodding their head in the direction of a 3D audio sound played through a set of headphones.

To evaluate the effectiveness of the using head and hand gestures as the input mechanism to the device, an experiment was conducted to access the usability of the interface. The authors measured the time to complete tasks, error rates and subjective workload while users where navigating through a realistic environment completing specified tasks. To access the impact on participants the average walking speed of users was also recorded for each experiment to measure the usability of an audio/gestural interface. Results from the two experiments carried out showed that walking speed was faster using hand gestures instead of head gestures and this might be explained by nodding of the head causing a slight pause in movement as the head looks downward. Overall users where able to walk at around 70% of their normal walking speed.

Their research shows that non-visual interaction techniques can be effective when in a mobile context that reduces the access to other modalities. These hypotheses can be extended from mobile situations to any situation where use of other modalities is either not appropriate or possible. By investigating more novel interaction techniques the effectiveness of interactions with devices/systems can be improved.

Brewster [11] recognise the effect that proactive computing systems will have on users interactions with them due to there large numbers and constant service availability. To avoid overburdening users and avoid the associated problems new interaction and control techniques are needed. Brewster states that a good starting point would be to study how people deal with complex situations by using our different senses to avoid any one being overloaded. As most interface design today is based around GUI's, what happens when user's visual attention cannot be dedicated to the GUI? Brewster states that new interactions are needed that take account of both input and output to overcome this problem.

The author evaluates the merits of using gestural interaction as an alternative input technique and the use of vibrotactile displays as an output technique. The advantage of using gesture is that visual attention is not required due to your kinaesthetic sense.

When your hands are full your head can still be used but care must be taken as some important issues, such as the recognition of normal movement gestures can be distinguished. He also states that other additional contextual information can be gleaned from sensors that can work alongside the gesture recognition to gather information on user activity. So unless we can design a system that is usable in all situations, then the system may become unusable.

Other possible output techniques must be for conveying information to users in similar situation as the input techniques discussed. Brewster discusses the merits of using a non-speech audio display and Brewster [12] showed how the use of audio to aid targeting for selection improved performance by 25%. He then goes on to discuss the use of vibrotactile displays for non-visual input. He shows there effectiveness in simple devices such as phones and PDA's in conveying a simple alert in certain situations. Related work in the area of tactile icons, also known as tactons, has shown that vibrotactile messages can be used effectively with audio and video, extending communication possibilities [13, 14].

Hinkley et al [15] investigated HCI issues when operating a handheld computer in a mobile context. They identify issues that a user of these devices must cope with such as performing everyday activities like walking, talking or driving while using these devices. Also the types of interactions that are performed with such devices in these situations tend to be in the regions of seconds or minutes and not hours. Given these characteristics the demands that these devices place on our cognitive and visual attention should also be minimised.

The authors believe that by augmenting mobile devices with sensors has the potential to address these issues. But what interaction techniques would benefit/problem from this approach and the implications for end users?

To test various interactions a two-axis accelerometer was attached to the Pocket PC. This gives the ability to sense tilt relative to gravity and also responds to linear acceleration. When attached to the Pocket PC it allows them to sense walking, looking at the display and holding it sideway. This sensed information is collected by a context server that acts as a broker between the sensors and the application.

Anytime the application needs to access this information it does so by polling the context server or by notification of an event. By implementing specific interaction techniques with users the authors hoped to uncover design and implementation issues and some preliminary user reactions.

There where 7 users selected for the test who where all described as using there own Pocket PC significantly, some even saying "*it controls my life*". To test a new interaction users where asked to activate the voice record functionality on there device. Traditionally to access this functionality a user would have to find the button on the device or activating a button on screen, which requires significant visual attention. However Hinkley et al [15] implemented the capability to hold the device in front of you and speak to record a message. When the user held the device in the position to record a small beep would be played to give early feedback that the gesture had been recognised and that the device has started recording. To explore their hypotheses that the sensed gesture for recording requires less visual and cognitive attention, the users where asked to record a memo while using a mouse in one hand to track an icon around a computer monitor. They also had to perform the recording task normally.

Results from the tests suggest that users found the sensed gesture, "*Quite a bit easier to use*" and could focus on what they where trying to do. They also commented that they would use this functionality more often if implemented with gestures. The authors performed a workflow analysis of the recording interfaces and found that the sensor based gesture required 3 less steps, while subjects felt "*concentration was required to find the button and then remember to maintain continuous tension on the button*".

The results from the user testing clearly show that the cognitive load placed on users can be reduced by providing different interaction techniques. The users also felt more comfortable with using the feature and said they would use it more often if gesture sensing was used. The interaction tested was specific to the device but does suggest that using gesture is an effective method of input when other modalities may be slower or unavailable.

Tactile Perception

[16] Produced a technical report on tactual sensing to access its suitability for providing feedback while driving a car. They found that the finger tips are the most receptive area of the body for spatial resolution [18]. The skin has two types of sensors, slow and fast adapting sensors. The rapid adapters RA are the ones of interest for vibration and have two different types, RA1 and RA2. RA1 are near the skin surface, have 3-4mm diameter receptive field and best respond to frequencies 10-60Hz. RA2 are deeper under the skin, have 20mm diameter receptive field so do not provide local stimuli and are exited in the 60-1KHz range.

[18] Found that these types of sensors are "*non-linear and time varying and their sensitivity varies with stimulus size, shape and duration*". These factors make any generalisation for particular situations difficult but [19] inferred the minimum physical size of a tactile array as 1cm square. [20] List some of disadvantages of touch, such as the susceptibility to habituation where the brain disregards a signal that is constant and the skin also adapts to become less sensitive.

Investigating Tactile Feedback

[14] First introduces the concept of Tactons as an additional method to communicate information to users instead of the traditional auditory and graphical feedback. Cutaneous perception is sensations such as pain, vibration and indentation that are picked up by the skins mechanoreceptors through tactile devices. Two basic types of tactile devices exist, one uses a pin array to stimulate the fingertip while the other uses point of contact stimulators that can be placed anywhere on the body.

The cutaneous sense can be a powerful method of receiving information. This can be seen by [21] who taught users a simple tactile language with 45 symbols, and they could recognise 38 wpm. [Tan and Pentland] used a 3x3 vibrotactile array attached to a persons back to give directional information.

Brewster et al [13] suggest that the use of the cutaneous sense to convey information is underutilised in HCI but suggests that it can be used to convey structured information to users. He says that, "*Tactons are structured, abstract messages that can be used to communicate complex concepts to users*" and he uses the analogy of an icon as a symbol that represents a concept but, "*tactons can represent complex interface concepts, objects and actions very concisely*". He goes on to explain how tactons differ from icons in the sense that an abstract mapping exists between the tacton and what it represents, so this needs to be learned.

Brewster then goes on to explain the fundamental design of tactons and groups them into three categories. Compound tactons represent an abstract mapping such as increasing the intensity to open a file and a decrease when closing a file. Hierarchical tactons consists of a node in a tree and each level of the tree inherits characteristics from the other. Transformation tactons are tactons that have various properties represented using a different tactile parameter.

Brown et al [14] produced a report into the effectiveness of Tactons. He recognises how the cutaenous sense can sometimes be overloaded and that complex information can be communicated through cutaenous sense alone, such as deaf/blind people using the Braille language. While the prevalence of vibrotactile feedback has increased today to become part of electronic devices, Brewster investigated how best to use this type of communication.

"*Tactons form a simple user language to represent concepts at the user interface*". He argues that before we can begin to use tactons we first need to identify what parameters can be used in encoding information through them. Apart from the basic parameters of frequency, amplitude, waveform and duration Brewster also investigates the use of "roughness" and "roughness and rhythm" to evaluate them to evaluate them as parameters.

Brown also argues that we cannot begin to design tactons before tactile parameters have been identified. The devices used in his experiments resonate at 250Hz and the applicability of the results may not apply to all devices with wider bandwidth. He states that frequency modulation alone is not a suitable parameter as skin has a usable range from 10Hz to 400Hz, unlike audio 20Hz -20000Hz. Amplitude as a parameter could be problematic as reducing it could degrade perception to make it undetectable, while an increase could cause pain [23]. Varying waveform is ruled out due to subtle differences being undetectable. Information however could be encoded by varying the duration of pulses.

Due to the limitations of the basic parameters just discussed Brown considers it necessary to consider more complex parameters to encode information with tactons. Rhythm is an important parameter in music but is equally important in tacton design and by grouping pulses of different durations together, rhythm can be created . Complex waveforms could also be used and are created by modulating two sinusoids to create perceptually different Tactons. Using spatial location of transducers to convey information has also been used but is not consider for experimentation by Brown.

Brown then carried out 2 experiments to test the different parameters just discussed. In the first experiment he tested the parameter of roughness perception by running 50 tests of differing pairs of stimuli and indicate which one felt rougher. The results indicated participant's perception of roughness increased as modulation frequency decreased. These results indicate that different stimuli can be distinguished but not whether it can be done so uniquely.

Leading on from the first experiment showing roughness as a parameter a second parameter was also used to convey more meaningful information, although tactons can be created using 1 parameter. His experiment used one tacton to convey two pieces of information to represent a received phone message. The type of message was represented by different rhythms for text, multimedia and voice, while the priority is represented by roughness and be low, medium or high. The average result for recognition of priority was 80% and type of message was 93%.

From the results of both experiments is can be seen that roughness and rhythm can be used to create tactons and that more than one type of information can be expressed in one tacton by varying these parameters. This makes it clear that information can be conveyed to user through the use of tactons.

Wearable Vibrotactile Displays

Tan and Petland [24] recognise that the practical challenge faced for wearable computing is the development of a display method. Visual displays are the most common but may not be practical for wearable computers. So wearable computers create a new challenge for tactual interfaces to exploit and they recognise certain challenges that must be overcome for the development of tactual displays. They must be lightweight to facilitate constant wearing ruling out any device which uses force-reflective displays, such as the EXos system worn by astronauts. Vibrotactile displays have a lightweight design due to the use of small vibration units and are easily powered. As tactual displays are relatively new compared to visual/audio Tan&Petland recommend that initially we use tactual displays as a supplementary display for wearable computers.

To investigate the usefulness [24] developed a lightweight tactual wearable display to test what users perceived when running the sensory saltation test [25]. The test is used to show how the pattern is perceived differently than the actual path the pattern follows. The display used was a two dimensional 3x3 array with equal spacing between the centres of 8cm. The sensory saltation test is usually carried out on a linear array of stimulators and is know to produce sensations such as the straightness of a line. By using their 3x3 array dimensions and running the same test the authors wished to see what different sensations could be derived.

The testing was carried out on 12 participants who where not made aware of the sensory salutation phenomenon. Most users reported a sensation of something crawling up their spine and reported sensation between the placement of the vibration units, which is consistent with the sensory saltation phenomenon. These results are interesting and show that some sort of interpolation is happening when the cutaneous senses are sensing the vibration pulses.

Lindeman et al [26] built and tested a 3x3 tactile array designed for use by soldiers in the field to provide vibrotactile feedback while wearing a head mounted display. The target environment for use placed certain design goals on the developed array such as being mobile, compact, and simplistic with low cost components. The control of the tactors was based on pulse width modulation (PWM) of the voltage by varying the amount of time each tactor (an individual unit in a tactile array) was activated for. They wanted to test various devices for use as tactors and test results using both to see if differing results where found. A series of tests was also planned to determine user's ability to sense location of stimuli, match tactors that are emitting the same intensity vibrations and the impact of vibrotactile cues in a search task.

To carry out their testing they developed the Tactorboard. The tactorboard is based on the Pic microcontroller and is attached to a computer via RS232 serial connection for communication with the board. From initial testing between Using cylindrical motors and coin type DC motors they decided on using the coin type motor as the tactors. Some interesting results from the testing of the DC motors show that when worn under a heavy backpack compared to sitting on the chest a higher voltage was required before user perceived the same level of intensity in the sensations. This would suggest that wearable display might have to be tailored to the conditions under which the wearable device expects to worn in for optimal use.

Three sets of testing was carried out, the first test showed that participants could recognise with 83% accuracy the correct location of stimuli, with some statistical error where users misidentified stimuli on higher back for lower. For the second test, to see if users could match tactors with the same intensity, results proved inconclusive and according to the authors show the need to change the frequency dynamically to counteract different loads on the tactors. The final study to investigate the affect of vibrotactile cues have in a visual task search showed that vibrotactile cue significantly reduced the time taken to complete the task. It still was not as significant as visual cues but still performed well.

Chapter 3: System Hardware Components

The project uses quite a lot and varied hardware components to build the complete integrated Gesture Recognition System with Vibrotactile feedback. The overall system could essentially be broken into two distinct sections, the Gesture Recognition System consisting of an MT9 [27], IPAQ 5500 [28], Total Remote (Audio IR Dongle) [29] and the Vibro-Tactile belt consisting of Keil MCBx51 [30] development board, four ULN2003AN [31] Darlington arrays and sixteen SAM A100 [32] vibration units.

3.1 High Level Component Design

Figure 3.1 shows the high level design of the overall system. The IPaq is the main processing unit and is where the main body of program code resides. The motion sensor is attached to a glove and worn on the hand of the user. Data is fed from the motion sensor to the IPaq for processing. The program on the IPaq will analyse the data stream arriving through serial port 1 and perform the gesture recognition on the data.

When a gesture is recognised the IPaq has two functions to complete. First it must play a sound file that represents the IR command for the recognised gesture, which causes the audio IR dongle to send the infra-red signal. Secondly the IPaq must send a character out through serial port 2, which is connected to the MCBx51 development board that controls the tactile array.

The embedded code on the MCBx51 runs in an infinite loop listening for data to arrive on its serial port. When data is received the board it will then find the display pattern that matches data received and the display the pattern to the user through the tactile array. The tactile array is worn around the user’s torso and is held in place by a belt.

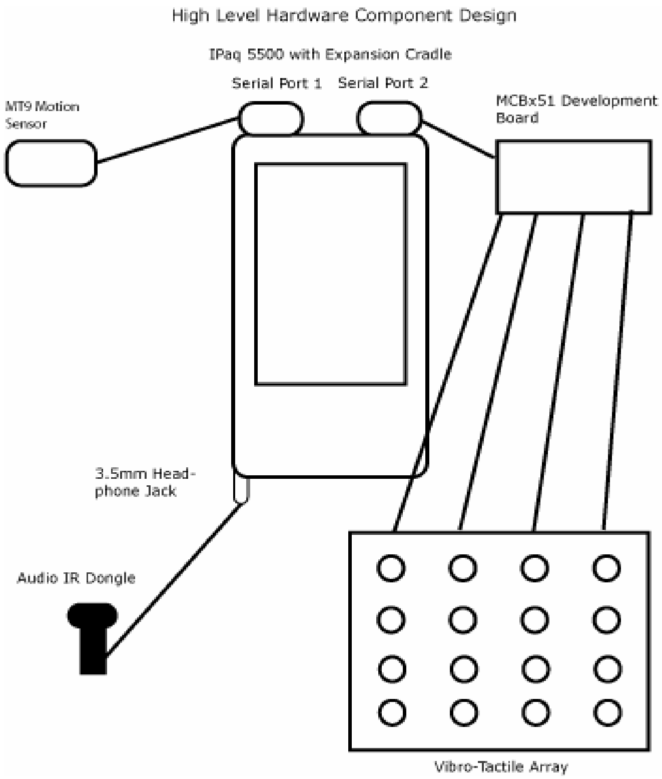


Figure 3.1 – Hardware Component Design

3.2 MT9 Overview

The motion sensor we used is the MT9-B from Xsens Technologies B.V. It is a small unit, 39mm x 54mm x 28mm which makes it slightly larger than a matchbox. The housing is ABS plastic and it is dustproof but not waterproof. It weighs only 35g making it ideal for our design. It is not a wireless device, but the connecting wire is flexible.

Figure 3.2 MT9 Sensor

The MT9 has a nominal power requirement of 6V and an operating temperature range of 0C to 55C supplied by the attached mains power adapter and it transmits data via an attached DB9 serial interface. The table below describes the MT9s internal sensors.

Accelerometers	- Solid State, capacitative readout
Gyroscopes	- Solid Stage, monolithic, beam structure, capacitive readout
Magnetometers	- Thin-film magneto resistive

3.2.1 MT9 Internal Sensors

The MT9 software contains a proprietary algorithm developed by Xsens tailor-made to the MT9 that can accurately calculate absolute orientation in three-dimensional space from miniature rate of turn sensors (gyroscopes), accelerometers and magnetometers in real-time. The design of the algorithm can be explained as a sensor fusion algorithm where the measurement of gravity (accelerometers) and magnetic north (magnetometers) compensate for otherwise unlimited increasing errors from the integration of rate of turn data. This type of drift compensation is often called attitude and heading referenced and such a system is often called an Attitude and Heading Reference System (AHRS). This means that for our purposes the MT9 can output absolute orientation information in three dimensions with no appreciable drift.

The MT9 can output the following orientation data:

3D Orientation as:

Quaternion (Roll, Pitch and Yaw)
Euler Angles
Rotation Matrix

Calibrated Data including:

3D Rate of Turn
3D Acceleration
3D Magnetic Field Strength
Temperature

Uncalibrated raw binary data:

The MT9 outputs it data via its DB9 serial connector at 155200 bps, 8 data bits, no parity and 1 stop bit. It does not use flow control. If raw calibrated data is required, the MT9 outputs at 100 Hz. If orientation data is required the MT9 outputs at 30 Hz. The immediate benefit here is that the MT9 performs the orientation calculation and sends data a lower rate to the IPaq therefore reducing the chance of overrunning the IPaq serial port.

3.3 IPAQ 5500

The overall goal of the system design was to finish with a wearable system that a user could walk around with. With this in mind I decided to use the IPaq 5500 to process the gesture data and act as the main processing unit of the system. The IPaq 5500 gives an advanced development environment along with a powerful 400 MHz Intel X-Scale processor in a portable unit. They are also readily available, as the computer science department has a number in its possession. In order to connect the IPaq to the 9-pin serial port on the motion sensor, I use a standard Brainboxes expansion dual-slot PCMCIA expansion sleeve [33], and a dual high-speed serial adapter card from Brainboxes Inc [34].

3.4 Total Remote IR Dongle

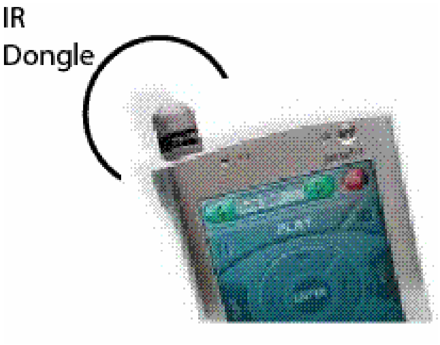


Figure 3.3 Griffin Total Remote IR Dongle

The overall system required the capability of sending an Infrared remote control signal to control the IR device of choice. One option was to use the IRDA port on the IPaq but this had a number of disadvantages. The first was the transmission range was 5 meters maximum, therefore limiting your control of device to anything within this range. The second was the incompatibility of using the IRDA port to send CIR (Consumer IR) signal involves hacking the IRDA port, although not impossible definitely not an optimal solution.

Griffin Technologies manufacture an audio IR dongle that converts an IR signal to an audio file. This file is then played out through the audio port of any suitable audio headphone jack. The IR dongle then converts the audio file back into an IR signal that has a transmission range of up to 100 feet with the correct audio hardware. The audio IR dongle comes with propriety software so a hack was required in order to integrate the dongle into the project, which will be explained later.

3.5 Keil MCBx51 Development Board

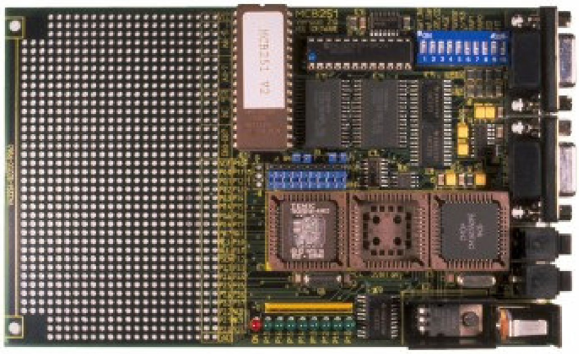


Figure 3.4 **MCBx51 Development Board**

When it was decided that the project would require a feedback mechanism for users I decided on a Vibro-Tactile array for this purpose. As nothing existed that could be bought for this specific purpose I would have to design build and integrate one into the system myself. Once I had decided on the vibrations units I was using for the belt I needed hardware built to control the units. Available in the college labs was the Keil MCBx51 development board.

The MCBx51 is a single board computer that supports the family of 8051 compatible devices. The Atmel 8051 microcontroller [9] is an 8-bit CPU and is widespreadly used in basic embedded control applications and was installed on the MCBx51. The Atmel 8051 microcontroller has 4 I/O ports that implement a quasi-bidirectional out put that is common on 8051's. Each of these ports has 8 I/O pins which are accessible on the prototyping area of the MCBx51 board, figure 3.4. This means that the ports can be used as both input and output without reconfiguration.

These quasi-bidirectional ports where something I was not familiar with when I initially planned on using the board and I assumed that once I connected my vibration units to the ports I would be able to control them directly with the microcontroller. This was an incorrect assumption and caused problems in controlling the belt initially and required the purchase of the ULN2003an Darlington arrays to overcome the problem, which I will explain later.

The MCBx51 also comes equipped with 2 on chip UART serial ports. One is for communication from a computer to the board for downloading code to flash RAM. The other is for communication between the microcontroller and another computer with standard RS232 communication. This is the port I would use to send messages to the board to control the Vibro-Tactile array.

There is a perforated breadboard, figure 3.4, extended onto the MCBx51 board for prototyping your own hardware. This was not planned to be used initially but was required to attach the ULN2003an too. The MCBx51 board also has lots of other features that are not used by me in this project. The board itself I would describe as a heavyweight solution but one I was semi-familiar with and had access to in the college lab.

3.6 ULN2003AN Darlington Array

As mentioned earlier the use of the ULN2003AN chip was not in the original design and was needed to use the quasi-bidirectional I/O ports of the Atmel 8051 microcontroller. The ULN2003An is a general purpose array that can be used with TTL and CMOS technologies. It is designed to operate around the voltage range of 5 volts, which is the same as the MCBx51 development board. The problem with connecting the vibration units directly to the I/O ports of the development board was that the units where drawing too much current from the port causing it to act as an output port. The effect of this was that I could not directly control the units, hence the need for the ULN2003AN chip. By connecting the ULN2003AN chip directly to the port pins and then connecting the vibration units to the ULN2003AN chip allows the chip to control the vibration units via the I/O port pins, figure 3.5.

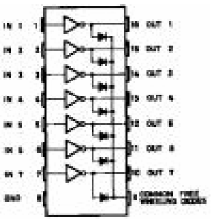


Figure 3.5 ULN2003an

3.7 Vibration Units

The choice of the SAM A100 vibration was based on the need for a coin type vibration unit that I wanted to use as the tactons for my array. There where other types of units available, mainly cylindrical and offset motors[10], but these simple mobile phone vibrations units where simple and cheap (3 Euro) to purchase. The smaller the diameter of the unit the more expensive they become. If it turned out that these where unsuitable then no expensive outlay was wasted.

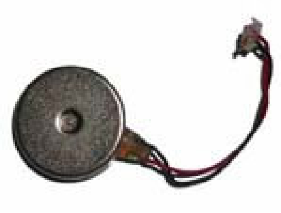


Figure 3.6 Vibration Unit

Chapter 4: System Analyses and Design

With the hardware components defined and the project research goal defined; to design and build a gesture recognition system with vibro-tactile feedback. The analyses and design of the system will now be explained. The techniques I used throughout the project was an object oriented design coupled with rapid prototyping development to ensure no time was lost on the design and coding of unsuitable algorithms for gesture recognition.

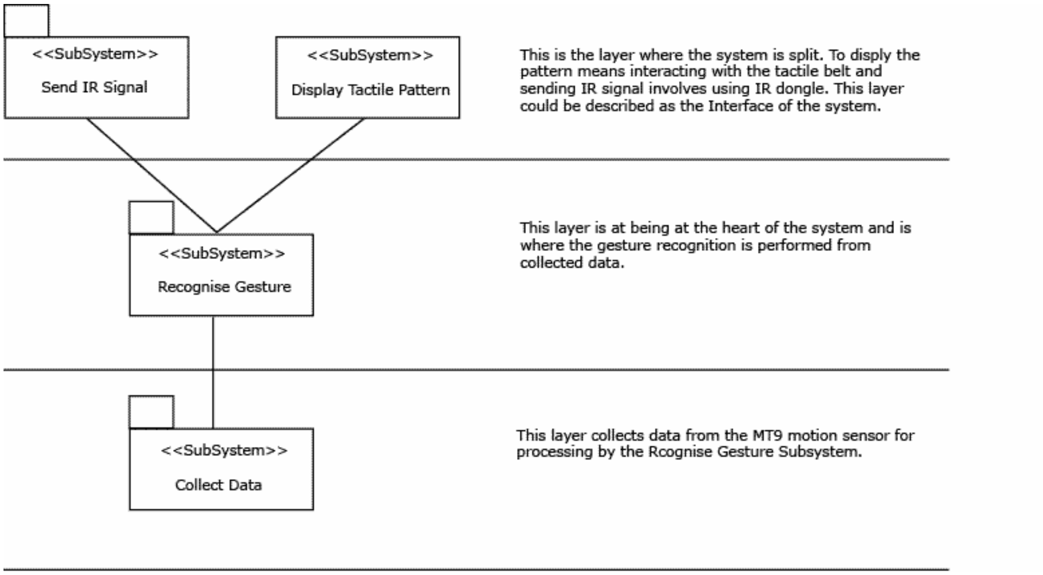
I will start this chapter by giving an overall system design to uncover the components of the system to be developed separately. From a coding and development perspective the project took two separate development approaches, with one predominately the gesture recognition software based around the data from the MT9 motion sensor, the other a hardware based tactile array controlled by the outcome of the gesture recognition engine. As the initial analysis of the project will show, it essentially involved two distinct components, which interoperate together. The analysis and design of both will be explained separately.

4.1 Overall System Analysis & Design

I will begin the by giving a description of the system and then develop some high level abstractions of the architecture of the system and go on to define the subsystems from this hierarchy. The analysis and design presented here can be thought of as the final or derived set of architectures and design, refined from many iterations of the processes covered.

Brief System Description

The system is a gesture recognition control system with a tactile array as the feedback mechanism. The MT9 motion sensor is attached to a glove worn by the user, the user will use hand/arm gestures to control a TV or similar IR controlled electronic device. Once the user performs the gesture and it is recognised by the system the system performs two operations. The first is to send the IR command to the device that is being controlled and the second is to display the pattern on the tactile array that is worn around the torso of the user. This is described in a layered architecture below.



<<SubSystem>>
Send IR Signal
<<SubSystem>>
Display Tactile Pattern
This is the layer where the system is split. To disply the pattern means interacting with the tactile belt and sending IR signal involves using IR dongle. This layer could be described as the Interface of the system.
<<SubSystem>>
Recognise Gesture
This layer is at being at the heart of the system and is where the gesture recognition is performed from collected data.
<<SubSystem>>
Collect Data
This layer collects data from the MT9 motion sensor for processing by the Rcognise Gesture Subsystem.

Figure 4.1 Layered Architecture of Integrated System

I think a layered architecture for the system is appropriate as each layer in the architecture is dependant upon processing from the lower layers. Note how the subsystem “*Display Tactile Pattern*” is not described in the conventional UML format due to this subsystem not representing a layer containing objects that carry out this function, but is comprised of embedded code on the microcontroller. The rest of this chapter will describe the design of these subsystems in detail starting with the most substantial and important, the gesture recognition subsystem.

4.2 Design of Gesture Recognition System

The gesture recognition subsystem function is to collect the data received from the data collection subsystem for processing. The data received from the subsystem is in three different formats, roll, pitch and yaw Euler angles. It is from this data stream that the system must decide whether a gesture was recognised from the data by comparing the data stream with stored values representing the gestures. If a gesture is recognised then this must be passed on to the other two subsystems. The recognise gesture subsystem can now be broken down into a number of different components.

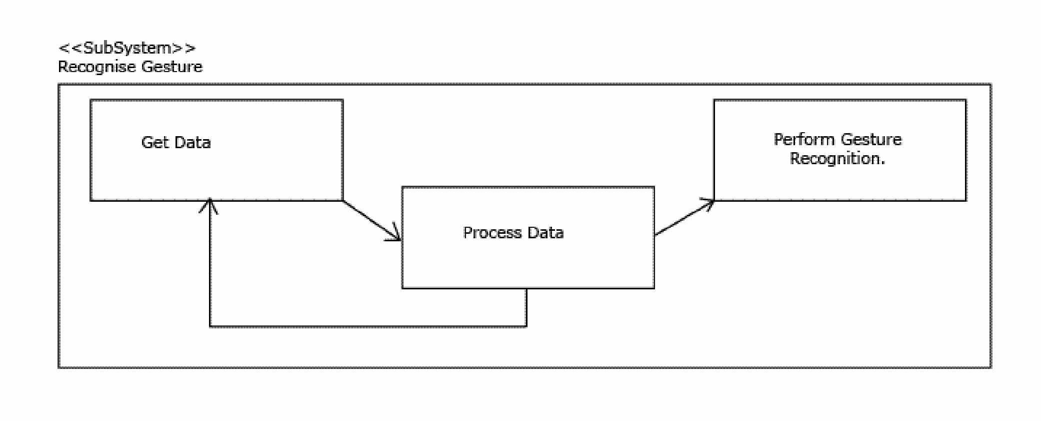


Figure 4.2 Recognise Gesture Subsystem

After this stage I had already formulated the basic objects in my design that would represent the object classes. Figure 4.2 shows the recognise gesture process first needs to collect its data for processing. This is a continuous process that never stops; the reason for the data processing before recognition is performed is to avoid running gesture recognition on a constant data stream, which has processing side affects for the hardware.

When the processed data is passed on for recognition, the system has recognised from the data stream the possible start of a gesture which must be tracked by the system to decide if a gesture was performed. By analysing the data stream of roll, pitch and yaw allows this decision to be made. Figure 4.3 is a UML diagram of the Gesture Recognition subsystem.

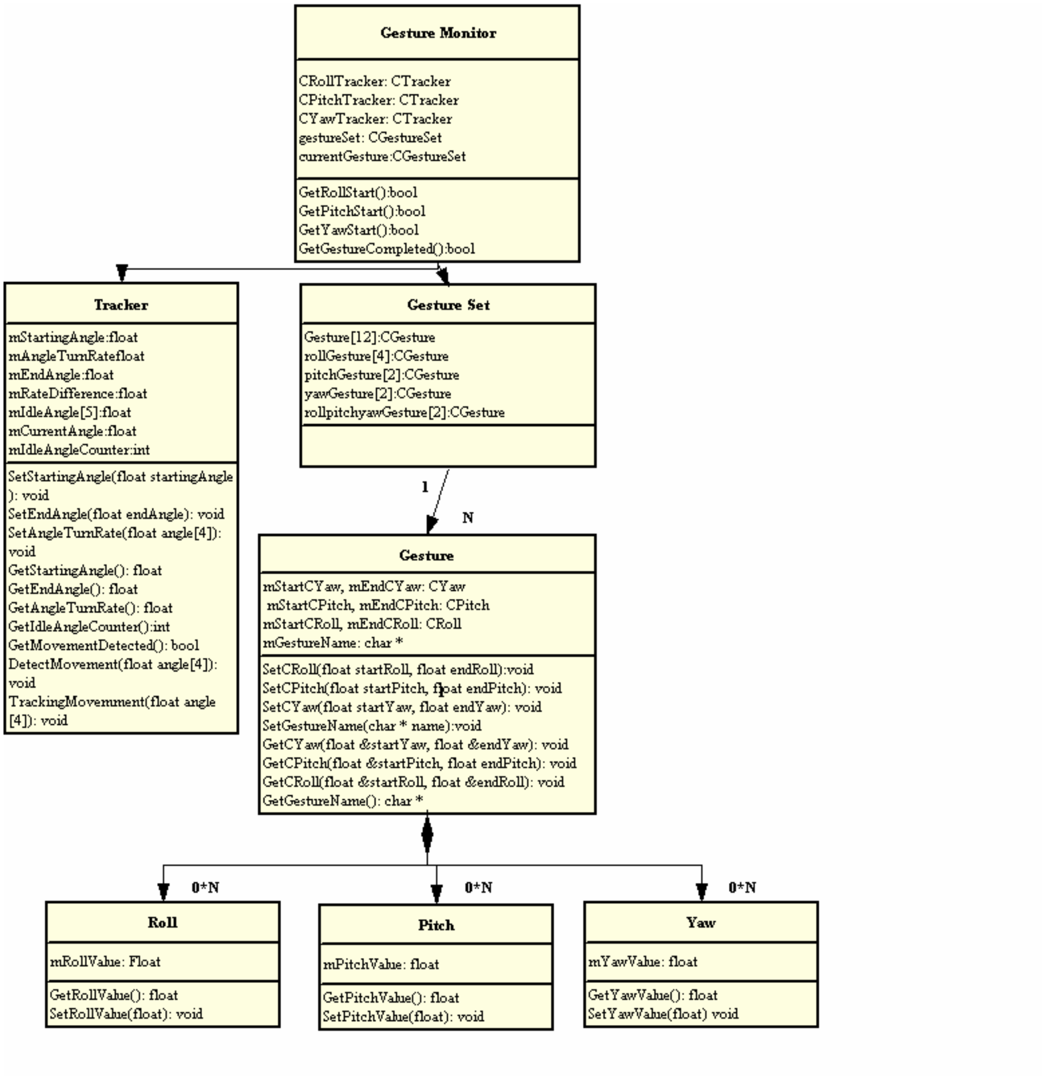

Gesture Monitor
CRollTracker: CTracker
CPitchTracker: CTracker
CYawTracker: CTracker
gestureSet: CGestureSet
currentGesture:CGestureSet
GetRollStart():bool
GetPitchStart():bool
GetYawStart():bool
GetGestureCompleted():bool
Tracker
mStartingAngle:float
mAngleTurnRatefloat
mEndAngle:float
mRateDifference:float
mIdleAngle[5]:float
mCurrentAngle:float
mIdleAngleCounter:int
SetStartingAngle(float startingAngle
): void
SetEndAngle(float endAngle): void
SetAngleTurnRate(float angle[4]):
void
GetStartingAngle(): float
GetEndAngle(): float
GetAngleTurnRate(): float
GetIdleAngleCounter():int
GetMovementDetected(): bool
DetectMovement(float angle[4]):
void
TrackingMovenmment(float angle
[4]): void
Gesture Set
Gesture[12]:CGesture
rollGesture[4]:CGesture
pitchGesture[2]:CGesture
yawGesture[2]:CGesture
rollpitchyawGesture[2]:CGesture
1
N
Gesture
mStartCYaw, mEndCYaw: CYaw
mStartCPitch, mEndCPitch: CPitch
mStartCRoll, mEndCRoll: CRoll
mGestureName: char *
SetCRoll(float startRoll, float endRoll):void
SetCPitch(float startPitch, float endPitch): void
SetCYaw(float startYaw, float endYaw): void
SetGestureName(char * name):void
GetCYaw(float &startYaw, float &endYaw): void
GetCPitch(float &startPitch, float endPitch): void
GetCRoll(float &startRoll, float &endRoll): void
GetGestureName(): char *
0*N
0*N
0*N
Roll
mRollValue: Float
GetRollValue(): float
SetRollValue(float): void
Pitch
mPitchValue: float
GetPitchValue(): float
SetPitchValue(float): void
Yaw
mYawValue: float
GetYawValue(): float
SetYawValue(float) void

Figure 4.3 UML Class Diagram of Gesture Subsystem

4.2.1 Explanation of UML Class Diagram

The GestureMonitor class is the controlling class of the gesture recognition system. It has 3 Tracker classes for each data stream and a GestureSet class that holds defined gestures that possible gestures are compared with. When the GestureMonitor class is tracking a possible gesture that data is then passed to the Tracker class which monitors the data stream to record when a gesture has started and ended. Once the tracking of all streams have finished the tracker class passes the recorded gesture back to the monitor class. The GestureMonitor class then checks the recorded gesture against the set of stored gestures in the GestureSet class to see if a valid gesture has been tracked and recorded.

The GestureSet class is made of many gesture classes which are possibly any combination of defined gestures for each data stream. So a Gesture class defines a gesture as a combination of at least one and possible 3 different classes of Roll, Pitch and Yaw gestures.

4.3 Design of Tactile Array

The tactile array is based on 16 (SAM A100) coin type vibrating units and controlled by an 8051 microcontroller attached to the Keil MCBx51 development board. The system will be connected to the Gesture Recognition software with an RS232 serial connection between an IPaq and the development board. The software on the board will run in a super loop waiting for a character to arrive into the UART buffers. The communication is implemented in only one direction as the board does not respond to any received messages back to the Gesture Recognition system.

Each received character corresponds to a pattern that must be displayed on the tactile array. The pattern to be displayed has a number of different temporal parameters, so the system must be capable of implementing real time delays in hardware when controlling the units. Below is example of CPU execution sequence.

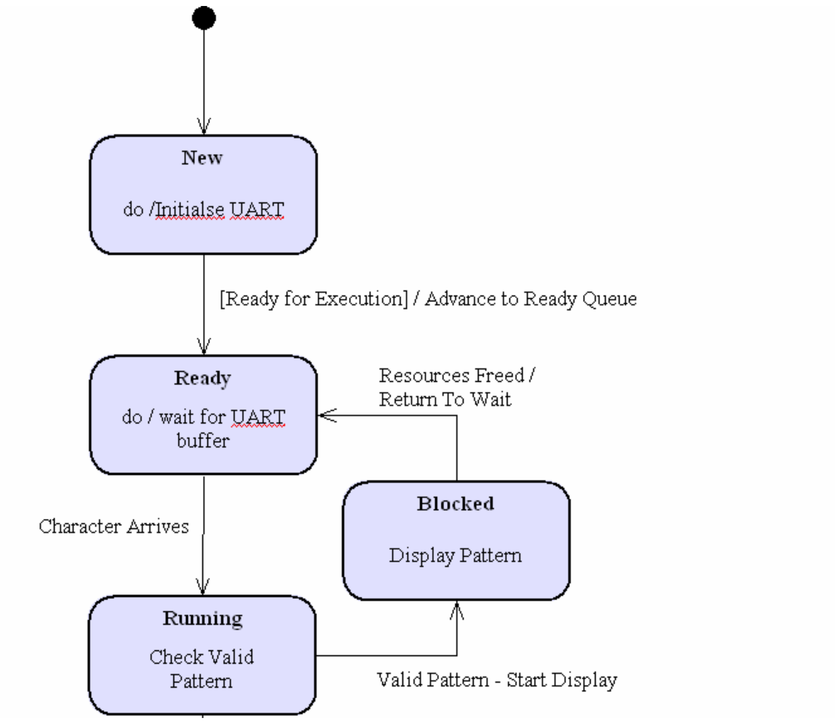

New
do /Initialse UART
[Ready for Execution] / Advance to Ready Queue
Ready
do / wait for UART buffer
Resources Freed / Return To Wait
Blocked
Display Pattern
Character Arrives
Running
Check Valid Pattern
Valid Pattern - Start Display

Figure 4.4 Embedded CPU cycle executions

From figure 4.4 it can be seen that the hardware must be able to perform UART communication with the gesture recognition system. It must also be able to display the tactile pattern which involves the control of I/O pins of the 8051 microcontroller. The system could be described as operating in soft real time as it does not have to meet stringent timing requirements.

4.4 Design of Collect Data Subsystem

Xsens already had a SDK for Visual C++ which I had planned as the basis of collecting the data from the MT9, so in a sense this module was already completed which speeds development time. The software for the MT9 is implemented in a COM-object (Component Object Model) [37]. All calls in software are then made via the MotionTracker object, see figure4.5. For the purpose of the project this functionality of the MT9 SDK dramatically reduced the time required to extract data from the MT9 sensors as all I had to learn was the required function calls to the Motion Tracker object from the SDK.

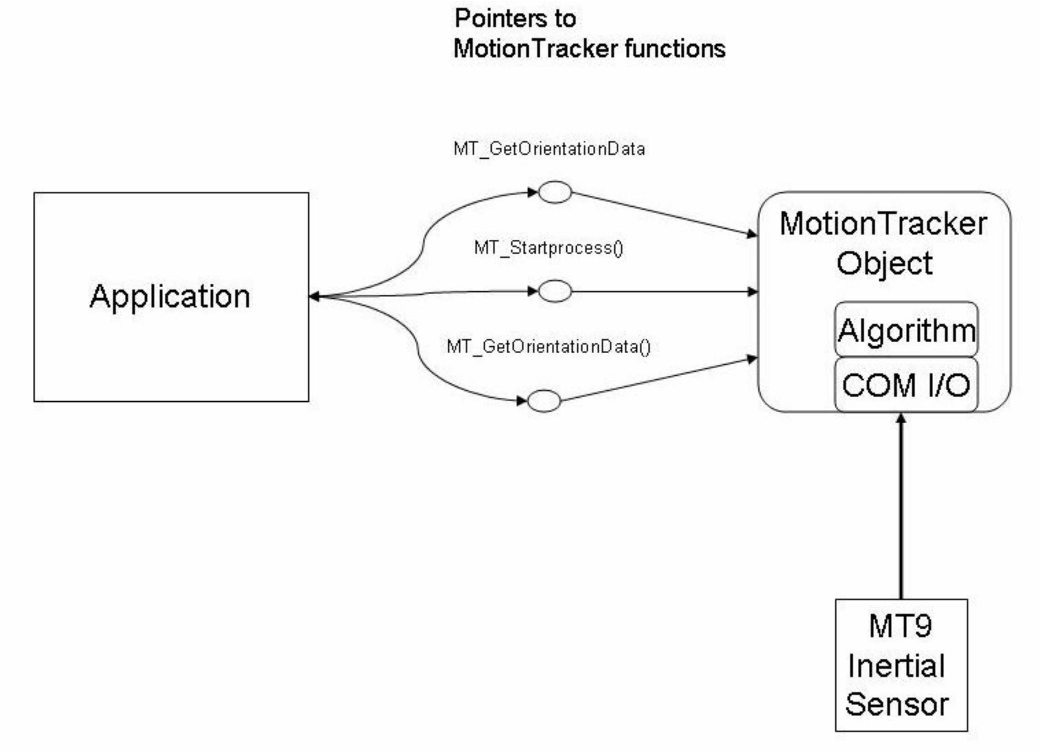


Figure 4.5 Motion Tracker Object

Design of IR Subsystem

The functionality of the IR system was to be provided by the Total Remote IR dongle that would allow me to play a sound file that would be converted to an IR signal through the audio dongle. There was quite some work involved in order to reproduce a sound file that when played produced the correct IR signal, which will be explained later. The use of the audio dongle avoids the other option of sending an IR signal, which is to perform a hack of the IRDA port on the IPaq.

Chapter 5: Gesture Recognition

In the previous chapter I explained the design of the gesture recognition system from system components through to a UML class diagram. This should have given an understanding of the components involved in the gesture recognition and their interactions. In this chapter I aim to show what I define as a gesture and how I recognise these gestures from the data stream. This involves showing how gesture parameters are defined and how the algorithm for the recognition operates. At the end of the chapter I will show how a gesture is recognised.

5.1 Definition of Gesture

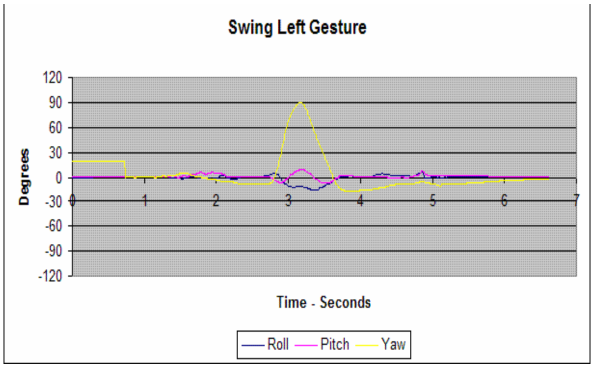
In the context of my dissertation a gesture is defined as any hand/arm movement that the system is trained to recognise. Each gesture starts and ends in the same position giving a space time curve that starts around zero and returns to zero for each data stream. This means that the system can be trained to look for these spikes in the data stream to start the recognition process. As the system is designed to act as a remote control of an electronic device the gestures where defined to have some metaphorical relationship with the command that the user was trying to accomplish. This gives a meaning to the gesture and should help the user remember and understand the gesture commands.

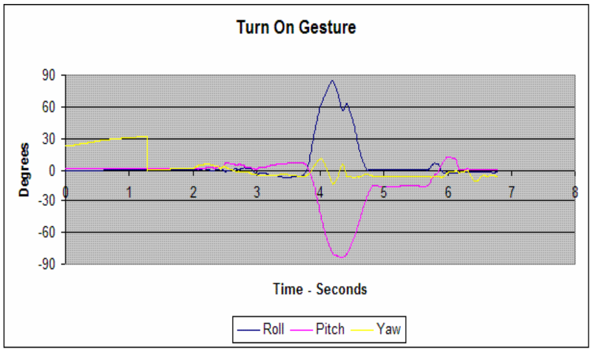
5.2 Sample Data Stream

Before I begin to go into the design of the gesture recognition algorithm, I will describe the data stream that feeds the system. The data shown is of 4 hand/arm based gestures that are where carried out by a user of the system. The data is generated by the MT9 motion sensor described in chapter 3 and consists of three data streams of roll, pitch and yaw. They are shown in figures 5.1.

I was given time to use the system and practice the gestures then performed each one individually. The streams shown demonstrate a gesture using each data stream individually (roll, pitch, yaw) and a gesture involving 2 data streams (roll, yaw). The data was collected from the MT9 at a sample frequency of 100 Hz and stored on file for later transformation into the graphs shown. Throughout the project I experimented with using different sampling rates from 50 Hz to 200 Hz but settled on using 100Hz through no use of empirical evidence as the most appropriate, but one which I got results with originally and stayed with.

As can be noted from the graphs shown, the gestures are a series of peaks in the data stream either above or below the zero baselines. The smooth peak and return to zero in the data stream is representative of the gestures preformed, as each is designed to start and return to the same relative position. Analysis of these graphs for performed gestures was the basis of the gesture recognition technique I employed. I would perform gestures and then analyse the data stream to see if the gesture could be recognised.





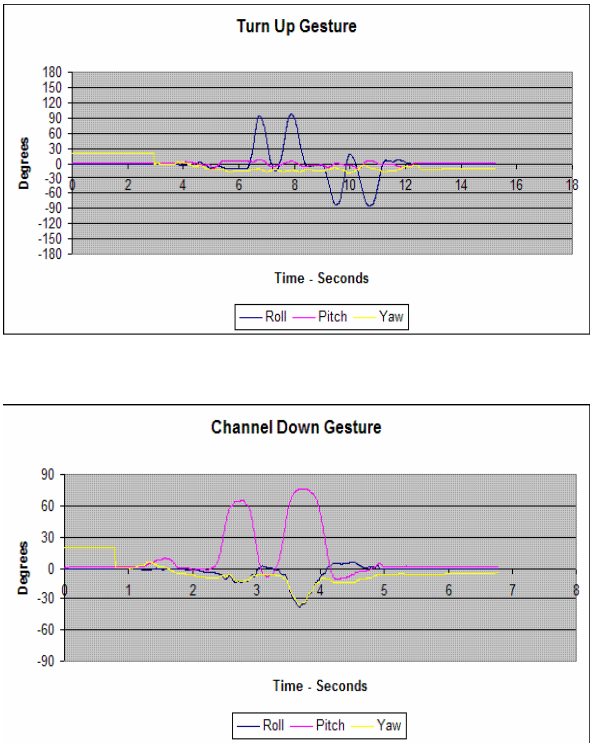


Figure 5.1 Example Data Streams

5.3 Pre Gesture Recognition Activity Detection

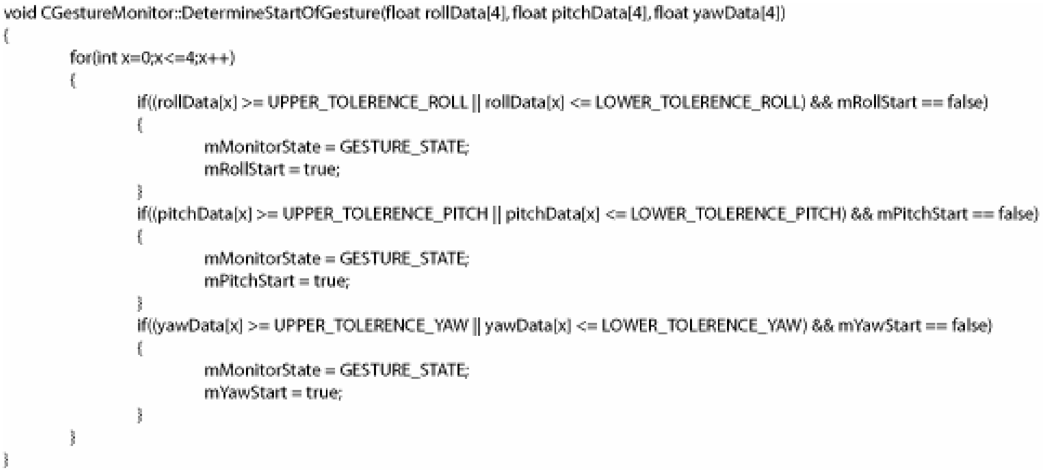
To cut down on the amount of processing any gesture recognition system has to perform requires the use of some form of activity detection [6] on the data streams in question. Without a mechanism to detect when someone is trying to perform a gesture the system must process the data stream continuously, which has consequences for the resources of any devices used. As the system is designed to be worn by users the devices must be kept as small and lightweight as possible to avoid making the system too bulky. The IPaq 5500, described in the chapter 3, was perfectly capable of processing the data stream on a continuous basis at the rates described in the previous section to recognise gestures. However this requires the use of an activity detection algorithm to monitor the data stream from the beginning of such gestures. Without this the system does not know where to begin looking in the data stream for gestures.

5.3.1 Pre Activity Detection

By restricting the gestures to start and end in a certain position allows me to monitor the data stream for this position. This is where the activity detection in the program starts, with data is collected from the MT9 and passed to the GestureMonitor class for further processing. When the GestureMonitor receives new data from the MT9 the monitor checks which state it is currently in and passes the data to the appropriate function.

Example:

On initialisation the system is configured as being in a sleep state and is not looking to start activity detection until the system data stream is within the parameters that indicated the start of a gesture.



```
void CGestureMonitor::DetermineStartOfGesture(float rollData[4], float pitchData[4], float yawData[4])
{
        for(int x=0;x<=4;x++)
        {
                if((rollData[x] >= UPPER_TOLERENCE_ROLL || rollData[x] <= LOWER_TOLERENCE_ROLL) && mRollStart == false)
                {
                        mMonitorState = GESTURE_STATE;
                        mRollStart = true;
                }
                if((pitchData[x] >= UPPER_TOLERENCE_PITCH || pitchData[x] <= LOWER_TOLERENCE_PITCH) && mPitchStart == false)
                {
                        mMonitorState = GESTURE_STATE;
                        mPitchStart = true;
                }
                if((yawData[x] >= UPPER_TOLERENCE_YAW || yawData[x] <= LOWER_TOLERENCE_YAW) && mYawStart == false)
                {
                        mMonitorState = GESTURE_STATE;
                        mYawStart = true;
                }
        }
}
```

Figure 5.2 – Determine Start of Gesture Function

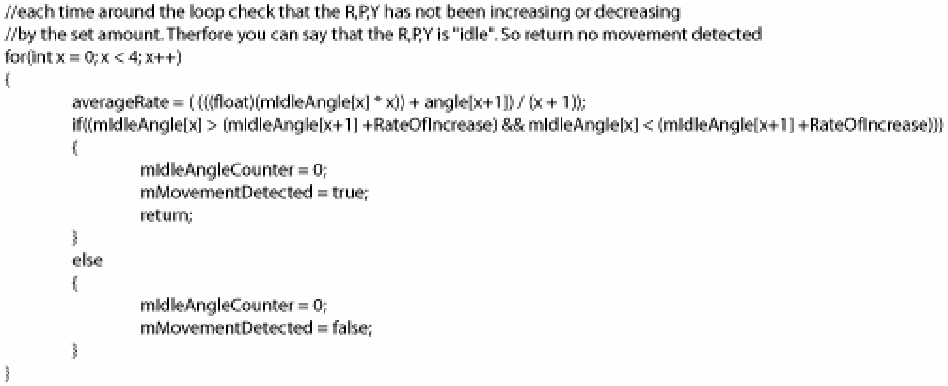
Figure 6.2 shows the function that the data from the MT9 is sent to before any activity tracking on the data stream is performed. The processing that has to be performed by the processor at this stage is a simple check to see if the data stream is within the bounds that I have defined as the starting point of every defined gesture. One the data stream comes within these bounds then the system will assume that the start of a gesture is imminent and the activity detection process starts from here.

As can be seen from figure 6.2 the roll, pitch and yaw data streams are passed through the system in an array of 5 values per function. The size of the array was determined from repeated testing to determine how many values where needed to give the program enough information to infer activity from the data. This was affected by the sampling rate which is 100Hz but with adjustments can work at the much lower rate of 50Hz but with a loss of resolution in the recorded data stream.

The advantage that is achieved when constraining the gestures into this position is that a user can use their hands to carry out normal activities without the system misrecognising gestures or processing the data stream. This simple constraint gives the activity detection a hook into the data stream without too much processing involved.

5.3.2 Activity Detection

At this point the system knows that the data stream is within the set bounds that a gesture begins from so now the system is looking for the start of a gesture. From here the system will now pass the data from the MT9 to the monitor which will now pass the data to a Tracker() class. The tracker class has two main functions to perform, the first is to detect the possible start of a gesture and the other is to track a gesture once it has been determined that it has started.



```
//each time around the loop check that the R,P,Y has not been increasing or decreasing
//by the set amount. Therfore you can say that the R,P,Y is "idle". So return no movement detected
for(int x = 0; x < 4; x++)
{
        averageRate = ( (((float)(mIdleAngle[x] * x)) + angle[x+1]) / (x + 1));
        if((mIdleAngle[x] > (mIdleAngle[x+1] +RateOfIncrease) && mIdleAngle[x] < (mIdleAngle[x+1] +RateOfIncrease)))
        {
                mIdleAngleCounter = 0;
                mMovementDetected = true;
                return;
        }
        else
        {
                mIdleAngleCounter = 0;
                mMovementDetected = false;
        }
}
```

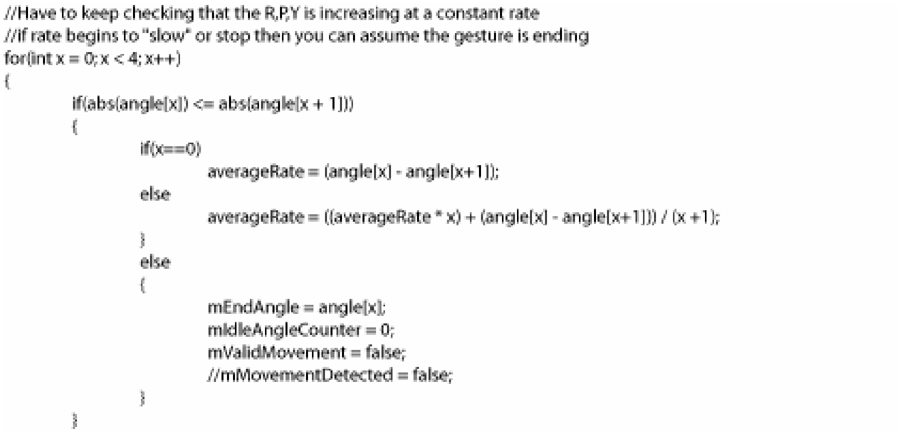
Figure 5.3 Detect Movement Function

The average rate of increase in the previous five samples are stored for five sample periods giving a time span of 250ms over which movement can be detected. The average rate of increase between 5 samples is cumulatively calculated and stored and gives the average rate of increase for a 250ms time period. This value is then compared against a predetermined rate of increase to determine if a gesture has started. This is currently set at 2 degrees for 250ms time period. Once this threshold is breached control of the program is passed on to track the data stream for an end to the gesture.

5.4 Gesture Tracking

At this point the tracker class can now process the data stream looking for the end of a gesture. In my original design I was monitoring the data stream looking for the data to indicate that the gesture had returned to the starting position. The parameters of the gesture at this stage where the point it started, the path it followed and the end point. Looking at the graphs in figure 6.1 show that the gestures consists of peaks, indicated by a curve that starts from zero and returns to zero. These types of graphs would be similar to what is graphed when using accelerometer [6] data, as the net integral over the time period is zero.

To improve the recognition times and to reduce the amount of processing required to recognise the gesture I refined my recognition process. By analyses of the data stream from using the graphs from 6.1 I hypothesised that the recognition could be carried out by recording the value of the peak instead of waiting for the gesture to return to the starting position. This also had an effect on the range of gestures that could be performed whilst cutting the recognition time in half.



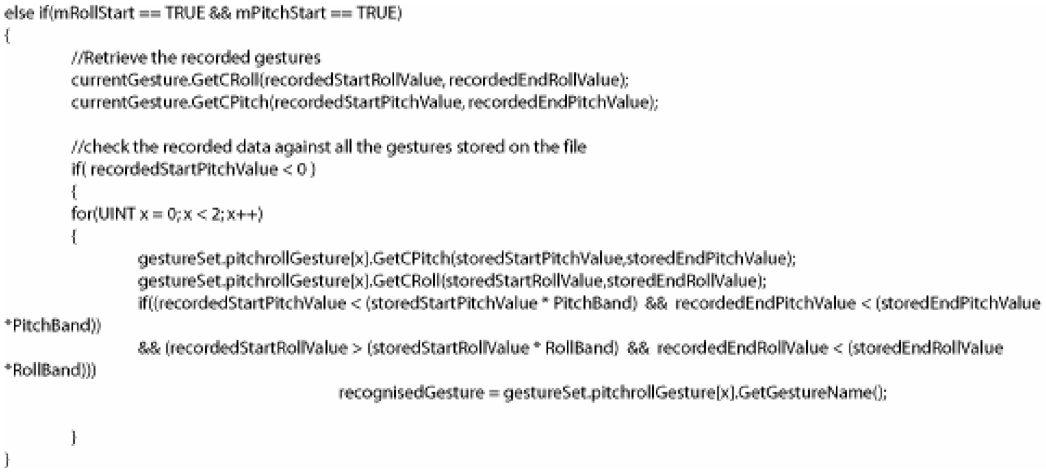
```
//Have to keep checking that the R,P,Y is increasing at a constant rate
//if rate begins to "slow" or stop then you can assume the gesture is ending
for(int x = 0; x < 4; x++)
{
        if(abs(angle[x]) <= abs(angle[x + 1]))
        {
                if(x==0)
                        averageRate = (angle[x] - angle[x+1]);
                else
                        averageRate = ((averageRate * x) + (angle[x] - angle[x+1])) / (x +1);
                }
                else
                {
                        mEndAngle = angle[x];
                        mIdleAngleCounter = 0;
                        mValidMovement = false;
                        //mMovementDetected = false;
                }
        }
```

Figure 5.4 Recording Gesture End Function Excerpt

Figure 6.4 shows an excerpt from the tracking gesture function that calculates the angular rate of increase on the data stream. This rate of increase is recorded for 50ms time slots for each data stream that is being tracked, which could be all three simultaneously. This rate is used in a comparison of a 250ms sliding window that is moved along the data stream with a push mechanism putting the last 50ms average in and discarding the oldest reading. These reading are compared to see if the average angular rate of increase is falling below the set threshold over the 250ms time period. When this condition is met the program assumes that we wave reached the end of our gesture and records the last value from the data stream as the end point of the gesture.

5.5 Gesture Recognition

At this point in the recognition process a possible gesture has been tracked from the start point, followed along its data stream until the gesture has been completed. Now the parameters of the tracked gesture must be compared with a GestureSet() class that contains the set of predefined gestures that can be recognised by the system. This gesture set was determined from performing gestures and recording their parameters. Anything that is tracked and compared against the gesture set but does not return a match is classified as an invalid gesture. Also because the program must track valid movement on each data stream it already knows what gesture set to compare the recorded gesture against. For instance if it is tracking the pitch and roll data stream then it only has to compare the gestures against the gesture set for these data streams.



```
else if(mRollStart == TRUE && mPitchStart == TRUE)
{
          //Retrieve the recorded gestures
          currentGesture.GetCRoll(recordedStartRollValue, recordedEndRollValue);
          currentGesture.GetCPitch(recordedStartPitchValue, recordedEndPitchValue);

          //check the recorded data against all the gestures stored on the file
          if( recordedStartPitchValue < 0 )
          {
          for(UINT x = 0; x < 2; x++)
          {
                    gestureSet.pitchrollGesture[x].GetCPitch(storedStartPitchValue,storedEndPitchValue);
                    gestureSet.pitchrollGesture[x].GetCRoll(storedStartRollValue,storedEndRollValue);
                    if((recordedStartPitchValue < (storedStartPitchValue * PitchBand)  &&  recordedEndPitchValue < (storedEndPitchValue
*PitchBand))
                    && (recordedStartRollValue > (storedStartRollValue * RollBand)  &&  recordedEndRollValue < (storedEndRollValue
*RollBand)))
                                        recognisedGesture = gestureSet.pitchrollGesture[x].GetGestureName();

          }
}
```

Figure 5.5 Gesture Comparison Function Excerpt

Figure 6.5 shows how the comparison is made to determine if the recorded gesture is a match of any of the recorded gestures. The gestures start and end points are checked to see if they are "*close*" to any of the gesture set. Closeness is determined by a quasi fuzzy logic comparison which is determined by the bands that are attached to each gesture. The bands set the bound on which the comparison must fall between. Adjusting this figure tightens the path that a gesture must follow to be recognised but decrease the likelihood of misrecognition of gestures, but an increase would have the opposite effect.

Chapter 6: Building the Vibrotactile Array

In chapter 4 I have given a brief overview of the execution sequence of the embedded software needed to drive the vibro-tactile array. This chapter aims to show how the array was built and explains the embedded code that was written to implement the design requirements.

6.1 Circuit Design

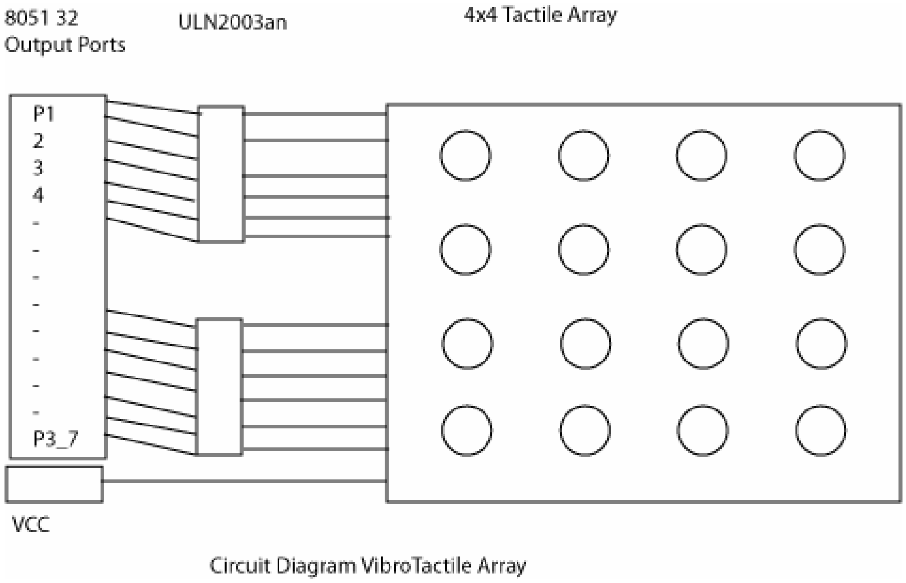


Figure 6.1 Circuit Diagram

When initially designing the vibrotactile array it was not envisioned that any extra components would be needed other that the Keil MCBx51 development board and the 16 vibration units. It was thought that the 32 output ports of the on board 8051 microcontroller would be enough to drive the units as needed and that the on board UART chip would allow for serial communication to the board. This meant that I did not have to build a complete IC from scratch and save in development time. However when the array was first built and connected to the output port pins of the onboard 8051, the current that coming from the pin was not enough to drives the units.

Initially I was unsure as to the reason for this and when using the multimeter to check the current I could see that when I connected the units the current was dropping to zero. I searched the 8051 forum on the internet and read the datasheet for the 8051 to see the reason the drop in current when trying to power the units.

The problem turned out to be my misunderstanding of how the output port pins of the 8051 are designed to operate. The data sheets described the output pins as being quasi-bidirectional. What this means that is that the output pins are designed to act as both input and output pins, depending on either the connecting component drawing or sinking current through the port. My units where drawing too much current and sending the pin low and forcing it to act as an output port which meant that I lost control of the port.

After searching through the 8051 forum I could see that this problem was not unique and could be rectified by using either relays or ULN2003 Darlington arrays. The advantage of using the Darlington array was the extra compactness compared to using a relay for each vibration unit. The Darlington's have 7 straight through input to output pins on one chip which allows me to control the tactile array using a minimum of three Darlington's.

After attaching the ULN203an chips to the development board, figure 6.2, I connected the vibration units to the output of the ULN2003an and then back to ground. This again resulted in the same problem I had originally, i.e. when I opened the port pins the current across the circuit was zero. I had not resolved the problem of the quasi-bidirectional nature of the port pins and they where being held low and not sourcing any current through the vibration units.

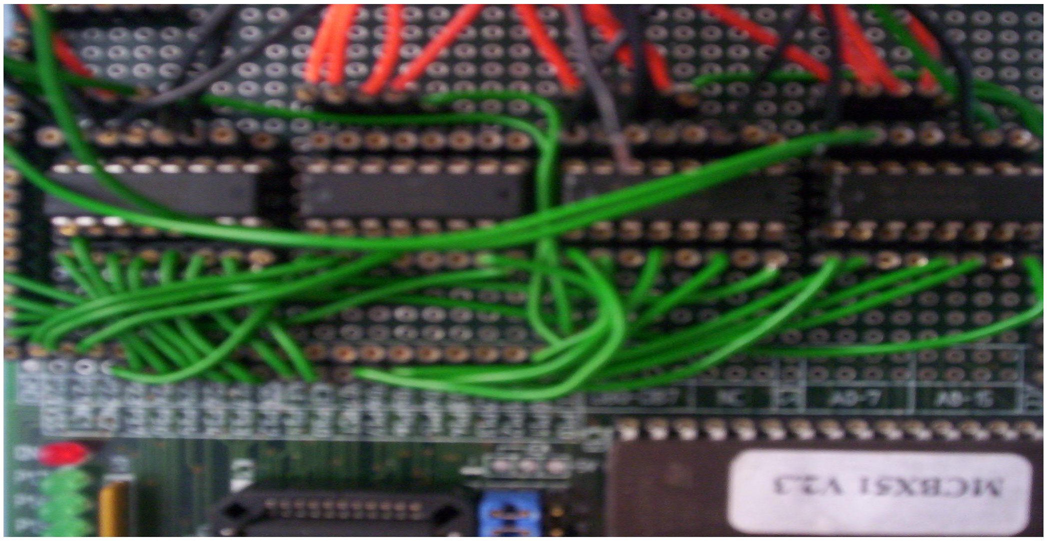


Figure 6.2 Darlington Arrays

After more searching on forums as to the cause to the problem I followed up a thread on one of the forums where a person was having the same problem using the Darlington's. One of the suggestions was to source the current from VCC through the units and sink the current down the port pins via the Darlington's, as in figure 6.1. This solved the problem and allowed me to control the vibration units by opening the port pins on the microcontroller which allowed the current flow to ground and completing the circuit.

Once I had control of the vibration units all the hardware for part project was completed. Now I had to write the software that would be used to control the units and the serial interface to the board allowing communication another computer.

6.2 Serial Communication

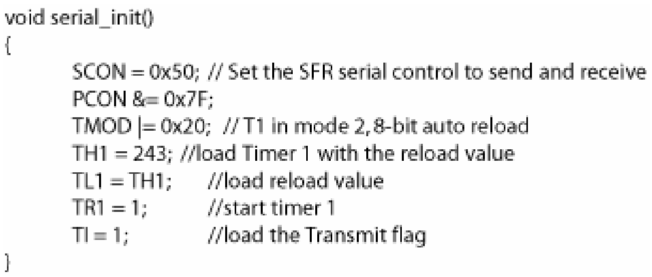
As the vibro-tactile array has to interface with the gesture recognition system a way of facilitating this communication was needed. The Keil development board already had an onboard serial RS232 connection so it was decided to use it as the communication mechanism. As the function of the array was to display patterns depending on what the gesture system desired, all that was needed was a simple mechanism to tell the board what pattern to display. This could be accomplished by a simple mapping from a character to a preset pattern. So all that was required to implement the communication was to use a one way serial line to the board to send a character down the serial line, which corresponds to the desired display pattern.

6.2.1 Implementing RS232 Communication

The 8051 microcontroller on board the Keil development board allows for serial communication to happen in two different ways. The first way is to clock your own transmissions for each bit that must be sent when writing a byte to the serial line. This is known as bare hands implementation and fortunately is not always required. Instead you can configure the serial port by setting the special function registers (SFR's) of the 8051, to configure the serial port to operate into the desired mode. After this is done the serial port baud rate must be set, then RS232 communication with the board can take place.

6.2.2 Setting the Serial Port Mode

To enable serial communication using the integrated serial port of the 8051 means setting serial control (SCON) SFR. This must be done to tell the microcontroller what baud rate we wish to use, how we will determine that baud rate and how many bits we will be using. Figure 6.3 below shows how to use configure the serial port mode for serial communication. The SCON SFR must be set to send and receive (=0x50) and the timer register TMOD must be set to the required mode of operation (|=0x20). Setting TMOD in mode 2, 8 bit auto reload means that timer low (TL1) value is reloaded into the timer high (TH1) register, when an interrupt is serviced by the overload of timer 1. This is how the baud rate is generated using timer 1 as the baud rate generator. The overload of timer 1 is set to correspond to the required baud rate for serial communication providing the clocking mechanism.



```
void serial_init()
{
        SCON = 0x50; // Set the SFR serial control to send and receive
        PCON &= 0x7F;
        TMOD |= 0x20;  // T1 in mode 2, 8-bit auto reload
        TH1 = 243; //load Timer 1 with the reload value
        TL1 = TH1;      //load reload value
        TR1 = 1;        //start timer 1
        TI = 1;         //load the Transmit flag
}
```

Figure 6.3 Setting the Serial Port Baud Rate

To determine the value that must be used as the reload value for the timer to generate the required baud rate involves a calculation that involves using two formulas. First of all you must calculate the reload value using the formula;

TH1= 65535 – (Oscillator Frequency / 32 x Baud Rate)

Once this has been calculated you plug the TH1 value into the second formula,

Baud Rate = (Oscillator Frequency / (32 x [65535 – (TH1)]))

The problem with using this formula with the Keil board was that the microcontroller was 12 MHz, which has the effect of not being able to recreate exact baud rates, a processor of 11.059 MHz is required for exact baud rates. Although any baud rate that is within 1% or better of the required baud rate will suffice.

6.2.3 Receiving a Character from the Serial Port

Once the serial port and timers have been configured serial communication is simple. The register RI is a one bit register that is set to 1 any time a character has been received and is in the serial buffer. To remove the character from the serial buffer is as simple as, *received_char = SBUF* . Then reset the flag and wait for another character to arrive.

Generating Display Patterns

The control of the vibration units achieved from earlier work was restricted to basic turn on/off control with no timing involved. As there where 16 units to be controlled from two 8 bit ports I mapped 8 units to each port. The units could be controlled by addressing the ports SFR or each unit could be mapped to a single bit port latch. By mapping the units to a port latch each pin could then be controlled by writing either a 1 or 0 to the port latch.

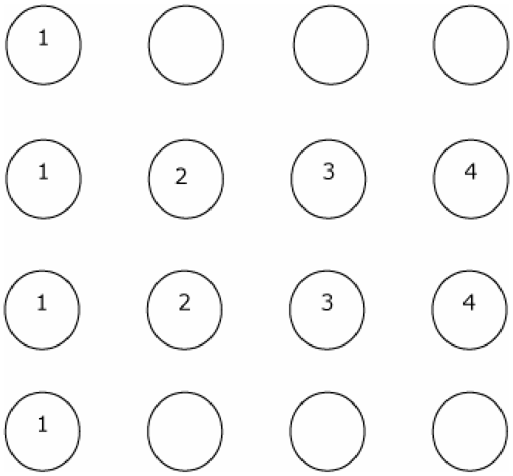
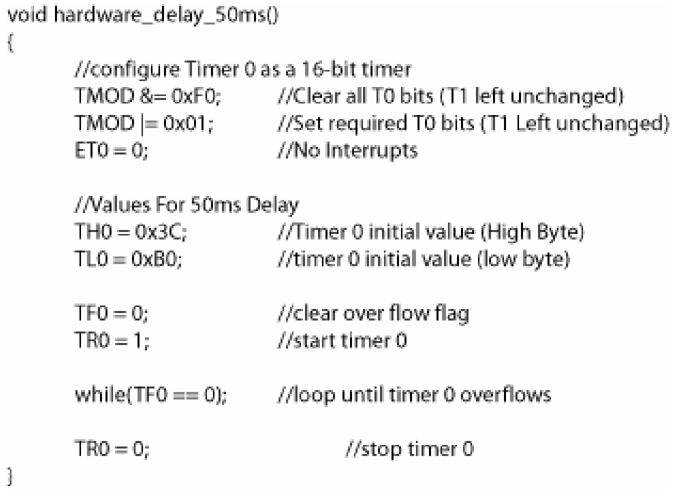


Figure 6.4 Left to right display pattern

To control the units requires a timing mechanism to control how long the units are activated for. Timer 1 was being used for the timing for RS232 serial communication which left timer 0 for creating a hardware delay for use in the displaying the patterns. Unlike the timing for the baud rate an exact hardware delay is possible. As timer 0 is also a 16 bit counter and with the 12 MHz processor updating the counter every 12 oscillator cycles. This means the timers are incremented 1 million times per second. So starting the timer at zero and waiting for the overflow flag to be set would indicate that 65.535 ms have passed. If I vary the start time then a delay from 1ms to 65.535 ms can be produced. The TH0 and TH1 values in the function below gives a 50ms delay that I can call in a loop to create delays that are a multiple of 50ms.



```
void hardware_delay_50ms()
{
        //configure Timer 0 as a 16-bit timer
        TMOD &= 0xF0;          //Clear all T0 bits (T1 left unchanged)
        TMOD |= 0x01;          //Set required T0 bits (T1 Left unchanged)
        ET0 = 0;               //No Interrupts

        //Values For 50ms Delay
        TH0 = 0x3C;            //Timer 0 initial value (High Byte)
        TL0 = 0xB0;            //timer 0 initial value (low byte)

        TF0 = 0;               //clear over flow flag
        TR0 = 1;               //start timer 0

        while(TF0 == 0);       //loop until timer 0 overflows

        TR0 = 0;                        //stop timer 0
}
```

Figure 6.5 Hardware Delay Function

To create the pattern above I would then turn on the required units, then start the hardware delay which gives the exact delay required. This process is then repeated until the entire pattern has been displayed.

Chapter 7: InfraRed Signal Generation

To control the electronic devices that the project was aimed at for testing and demonstration purposes required the ability to generate the IR signals for the device. Using an IPaq as the controller for the system gave me the option of using the onboard IRDA port. As I had previous experience of using this port to produce a CIR signal, for which it was not intended, I decided not to use this option due to the work that is required to hack the IRDA port for this purpose. Also the distance that an IRDA signal can travel is just 1 meter [38], which means standing in front of the device you wish to control for the IR signal to reach the receiver. This was why I decided to use the Griffin Total Remote which can generate a CIR signal that can operate at distances of up to 100 feet.

The problem with the Griffin Total Remote was that is was designed to operate with Griffin Total Remote software, so a hack was needed to make it work for the project. From initial investigations into using the total remote outside its intended purpose, it seemed that getting it to work as my IR device would be simple. This was not the case, as getting the dongle to work properly involved testing the device on at least 5 different IPaq's and various laptops before finding one that would operate it. This is due to problems with certain sound card hardware.

7.1 Generating an IR Signal

The griffin IR dongle is designed to be plugged into the 3.5mm headphone jack of any IPaq to act as a remote controllers IR emitter. To use your IPaq as a remote control you would record the IR signal from the remote you wish to emulate, using the total remote software, via the IRDA port. The total remote software then operates by converting this recorded signal to a sound file, that when played back trough the total remote dongle would generate the same recorded IR signal. What I needed from this process was the sound file that was created for the required IR signal. Once I had this

I could then playback this file in my program when wanting to control some device. As these where created on the fly by the total remote software on the IPaq I would need another way to extract them.

As the sound file was being played from the IPaq by the total remote software I could capture this by using a 3.5mm to 3.5mm headphone jack connector to route the sound to a PC and use sound forge to record it. Below is a picture of how the IR signal is represented as a sound file.

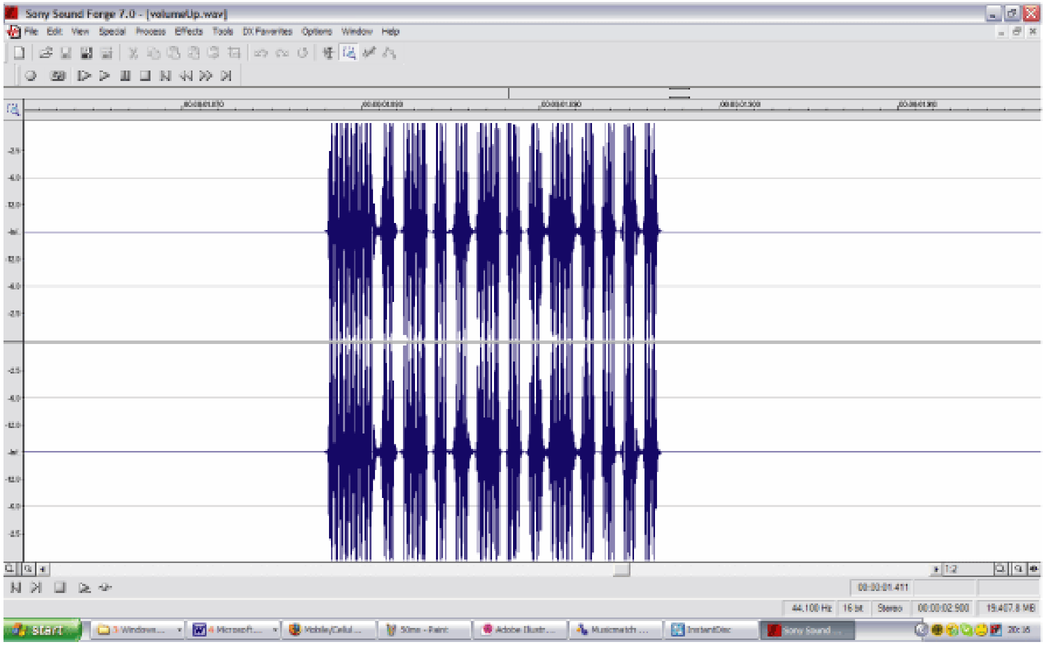


Figure 7.1 IR Signal

I repeated this process for every IR signal that I needed which meant that I now had the sound files for each required button on the remote control. Now I had the ability to use the IR dongle without using the griffin software by playing the sound files in my program when I wanted to operate the required device.

Chapter 8: Testing

8.1 Testing the Vibrotactile Array

The testing of the Vibro Tactile array was carried out using two distinct designs for the array. The first design and build of the array used 30mm to separate the tactons from centre to centre for each column and 20mm between the centres of each row. The second test sessions were carried out with 60mm separating the tactons for both the column and row. A redesign and build was needed due to unfavourable results obtained from the initial tests. The same test where used for both versions of the array.

Description of Users

The participants for the test were a combination of the colleagues from my Master's course and friends and family who visited my home. Before the test started each user was informed that it was the array that was being tested and not them. The format of the test was explained in detail before the test began to ensure all users were clear on what was being performed. The users were then given a questionnaire to fill for part 1 of the testing which they filled out as test progressed.

8.2 Description of User Tests

1) Pattern Recognition – Varying How Pattern Displayed

The purpose of the first test carried out on each individual was to determine the best method of displaying the pattern by varying the display method. The users were shown pictures of the patterns on paper which explained how the pattern was to be displayed. As the purpose of this test was to ascertain the best method of displaying a pattern it was decided that showing the patterns would not affect the test and would have the effect of familiarising the user with the patterns which would be tested in the second test set.

The first set of tests will be carried out by buzzing the patterns one row at a time or 1 tactor at a time, depending on the pattern. The second test will display the pattern in its entirety by turning on the tactors in the pattern simultaneously. The time taken to display the pattern for each method was equal. The constant display is displayed for 2 seconds then a break of 5 seconds, then displayed again for 2 seconds. The rolling pattern is displayed taking 2 seconds from start to finish. There is a pause of 5 seconds and then the pattern is displayed again.

There were a total of ten patterns to display with two methods for each pattern. After each round the users were asked,

	Constant	Rolling
1. Was the pattern displayed easily distinguished	(y)(n)	(y)(n)
Preference -		

The preference question was explained to the users. This question was to determine from a usability perspective which method "felt" better. What I mean here by "felt" was a general reaction after the pattern was displayed as I was trying to determine how comfortable using the belt felt. The users could also choose neither method, signalling that the experience was an uncomfortable one.

2) Pattern Recognition – Using Temporal Variations

These sets of tests will be used to find the best recognition rates for the various patterns that are displayed by varying the time each Tacton is displayed and the time between (gaps) each displayed tacton. By using these variations and reviewing the recognition results I was hoping to reveal the best set of parameters for displaying the pattern. I will also vary how long the pattern is displayed as either constant or rolling.

While performing these tests the users where not shown a description of the pattern that would be displayed before each test. This would allow me to document as the tests precede what the recognition rates are using by varying the different parameters of time and display type. As the users where now already familiar with the patterns from the first test I could reliably test there recognition of the patterns by randomly choosing which pattern to display and ask them for there answers. They could still view the pattern sheet if they wished to do so to reduce the memory load of trying to remember the layout of each one.

The test involves me displaying a pattern on the array and asking the users to state what they think the pattern displayed was. I would record whether their answer was correct on my question sheet for each test giving me the data I need to analyse later.

3) Pattern Recognition – Range of Patterns Displayed

The purpose of the first two tests was to derive the best display pattern technique, either constant or rolling and the best temporal parameters that give the highest recognition rates. With this defined I could then use these parameters to test different display patterns to try and discern the range of patterns that could possibly be recognised.

After the first two tests where completed I then picked a range of patterns to be displayed to try to distinguish exactly how many patterns users can recognise. I will allow the users a training period to get familiar with the patterns, and then run the display tests randomly. The format here is basically the same as the first test but with a determined set of parameters derived from analyses from the first two sets of tests.

8.3 Feedback System for Remote Control

Here users will be given the Gesture based Remote Control to operate a television or stereo. The users will receive feedback through the tactile display on their lower back that is mapped to the required gesture. The purpose of the testing here is to gain an insight from user testing if users consider the feedback for the system appropriate.

8.3.1 Analyses of testing on Array 1 Design

The first set of testing was carried out using an array with dimensions of 30mm to separate the tactons from centre to centre for each column and 20mm between the centres of each row. Overall the recognition results where very low with 0% recognition using the constant pattern display and 10% recognition using the rolling display. The temporal variations had no impact on the recognition rates which would suggest that the design was incorrect as users where obviously having trouble discerning any real sense of the displayed patterns. Users also overwhelmingly said they preferred the rolling display as opposed to the constant which suggests that they could not perceive any pattern from the constant display.

Analysis of Test Data

Test 1 - Pattern Recognition – Varying How Pattern Displayed
This test was to discover the best technique for displaying the patterns between rolling and constant. The tests where carried out by 12 users, 10 of these where fellow M.Sc. Computer Science students with the 2 remaining users non Computer Science students. The testing was carried out over a two day period in the computer science labs in Westland Square and in my home also.

As described earlier the users where shown diagrams of the pattern to be displayed before each test, then the pattern would be displayed using each technique and asked if they could recognise it. The recognition for the constant display was 0% for all users tested and 22% for the rolling display with all users selecting the rolling display as there preferred technique.

As the results suggest user recognition and satisfaction with the rolling display would indicate that the technique was not good for displaying tactile patterns as no user could discern any pattern from the individual test patterns. Also no user selected the constant display as their preferred type, even when no technique produced a positive recognition result. When verbally questioning the user afterwards to gain feedback they all indicated that the constant display felt like the whole belt was vibrating and had no awareness of where on the belt the tactons where vibrating. Whereas users indicated they could get a general sense of where on the belt the tactons where being activated using the rolling display.

Looking more closely at the data to see the patterns where users gave a positive response to the questionnaire reveals only on patterns (*Appendix 1*) 3, 4, 5 and 6 received any positive response. Patterns 3 and 4 are left to right and right to left displays respectively and 5 and 6 are Clockwise and Anti-Clockwise patterns. As was mentioned earlier the dimensions used for this test was 30mm to separate the tactons from centre to centre for each column and 20mm between the centres of each row. This means that there was a bigger gap (physically) between the tacton columns which patterns 3 and 4 where rolling across and also the patterns for 5 and 6 where spaced further apart than any other pattern. I hypothesised from this that spacing the tactons further apart might increase my recognition rates by allowing more distance between the tactons.

Test 2 - Pattern Recognition – Using Temporal Variations

As described earlier the purpose of the second set of tests was to determine the best parameters to use for displaying the patterns by varying the temporal parameters such as the duration of time the tactons vibrate for, the duration of the silence period between pulses therefore the total time taken to display the pattern. For this test the users where given the patterns at random and asked to say which they thought it was upon which I recorded their answer. They where given a sheet that had diagrams of all patterns printed on them for the users to use as a reference when making there decision and where not time limited to answer.

Three sets of different parameters where used,

1. On 250ms Off 250ms for each pattern using a rolling display.
2. On 250ms off 0ms for each pattern using a rolling display.
3. On for 2 seconds using the constant display.
4. On for 1 second using the constant display

As could be predicted from the first set of test results the recognition rate was only 12%. The users where asked not to make a guess and state that they could not recognise a pattern if this was the case to avoid skewing the results with guesses. Again, as in the first test the only positive recognition was with patterns 3, 4, 5 and 6 using a rolling display. The variations used in timing did not have any effect on the recognition rates.

8.4 Conclusion of Initial Testing

The recognition rates achieved meant that the third set of testing would be postponed as there was no point proceeding with these test cases. To carry out these tests I needed to have higher recognition rates and a derived set of parameters to use as exemplars. As mentioned earlier I had reasoned from the test data and personal observations that the tactors where too close together to discern any displayed pattern. This would seem to be confirmed by [20] where susceptibility to habituation occurs when the brain disregards a signal that is constant and the skin also adapts to become less sensitive. So it was at this point I decided to redesign and build the tactile array again with bigger dimensions and compare the different testing results.

8.5 Testing of Redesigned Tactile Array

When a redesign was decided upon the first thing I had to decide was the dimensions of the new array. Taking into consideration hypothesis that the closeness of the tactons was causing the low recognition rates I decided that a doubling of the array size to 60mm separation of the tactors for both row and columns. I decided that I should double the size of the array for various reasons, the main one being the time of dismantling the array and reassembling it takes time. This meant that small increments and retesting would have taken too much time. I decided upon doubling the size to see if my hypothesis that the close proximity was causing the problems the user where having with recognition of the patterns.

Test 1 - Pattern Recognition – Varying How Pattern Displayed

The format for the test was exactly the same as earlier in section 8.3 and the users where again my colleagues and friends. The result was exactly the same for the constant display type with all users not able to distinguish any display pattern. This would indicate that the constant display type was not suitable design for conveying any display pattern to the users with the design and components I had used. Upon asking users why they thought they could not distinguish any pattern, the most common response was that the whole belt was vibrating and nothing was distinguishable.

The results for the rolling display was much better and recognition rates where at the level where previous research projects[][][] had reached with 100% of users clearly able to distinguish patterns 1 to 6 and 50% recognition of patterns 7 and 8 which are a constant display type only. The results would seem to indicate that the spacing of the tactons was vital to achieving any recognition. The closeness could have perhaps been for the different display narrowed with better quality materials used to build the array and the tacton components. By reducing the physical size of the tactors and possibly being able to control the current and therefore the intensity of the vibration, greater testing could be done by varying these extra parameters to see what is achievable with the array sizes. If I had the option of reducing the intensity of vibration through the tactors better results using the constant display might have been achieved.

Test 2 - Pattern Recognition – Using Temporal Variations

Once again the format for the test was exactly the same as section 8.2 with colleagues and friends as the test subjects. The test where carried out to determine the best temporal parameters to use for the rolling and constant display techniques. However due to time constraints and the conclusive results from test 1 I decided not to perform any tests on the constant display pattern and concentrate on achieving the best parameters for displaying the rolling pattern. The data uncovered here will help when deciding how to best display the patterns when in use with the gesture recognition system as closing the loop from gesture recognition to feedback through the tactile array will be time constrained to ensure good usability.

The tests where carried out with varying temporal parameters and by varying the length of time taken to complete the display, either 1 or 2 seconds and varying the length of time of tacton pulses and length of silence period between pulses.

As can be seen from the figures the best results where achieved at 500ms pulses of each row or column of tactors which decreases to nearly unrecognisable at 250ms pulses. Above the 500ms pulse there was no increase in recognition levels which peaks around the 500ms mark. The tests also show that a delay time between the pulses did not improve recognition at the higher levels but did improve recognition at the lower 250ms level. The advantage of an improve rate with the added delay at lower levels is lost due to the added silence delay giving the same total time to complete the pattern as a higher rate with no delay.

8.6 Conclusions of Testing

The first test set was emphatic, the rolling display was recognisable by users but the constant display was unrecognisable. The effect of this is that the array would be limited to display patterns that could be displayed using the rolling format as trying to display a constant pattern would be unrecognisable. This would limit any patterns designed to work with the gesture recognition system as trying to display patterns in the same physical shape as the gesture may not be possible. Instead a display that was metaphorically the same as the meaning of the gesture would be more appropriate or a simple learned mapping between tactons and gestures.

The recognition of the display patterns for test set 2 provided the data for recognition rates for the different temporal parameters. Pulses above 500ms produced no increase in the recognition rates and anything below 250ms was unrecognisable. The test also revealed that adding a silence delay between the pulses had no effect at the higher end but helped at the lower 250ms range.

After completing the tests I now had the necessary data that would allow me to tweak the array when integrating it as the feedback mechanism for the gesture recognition system.

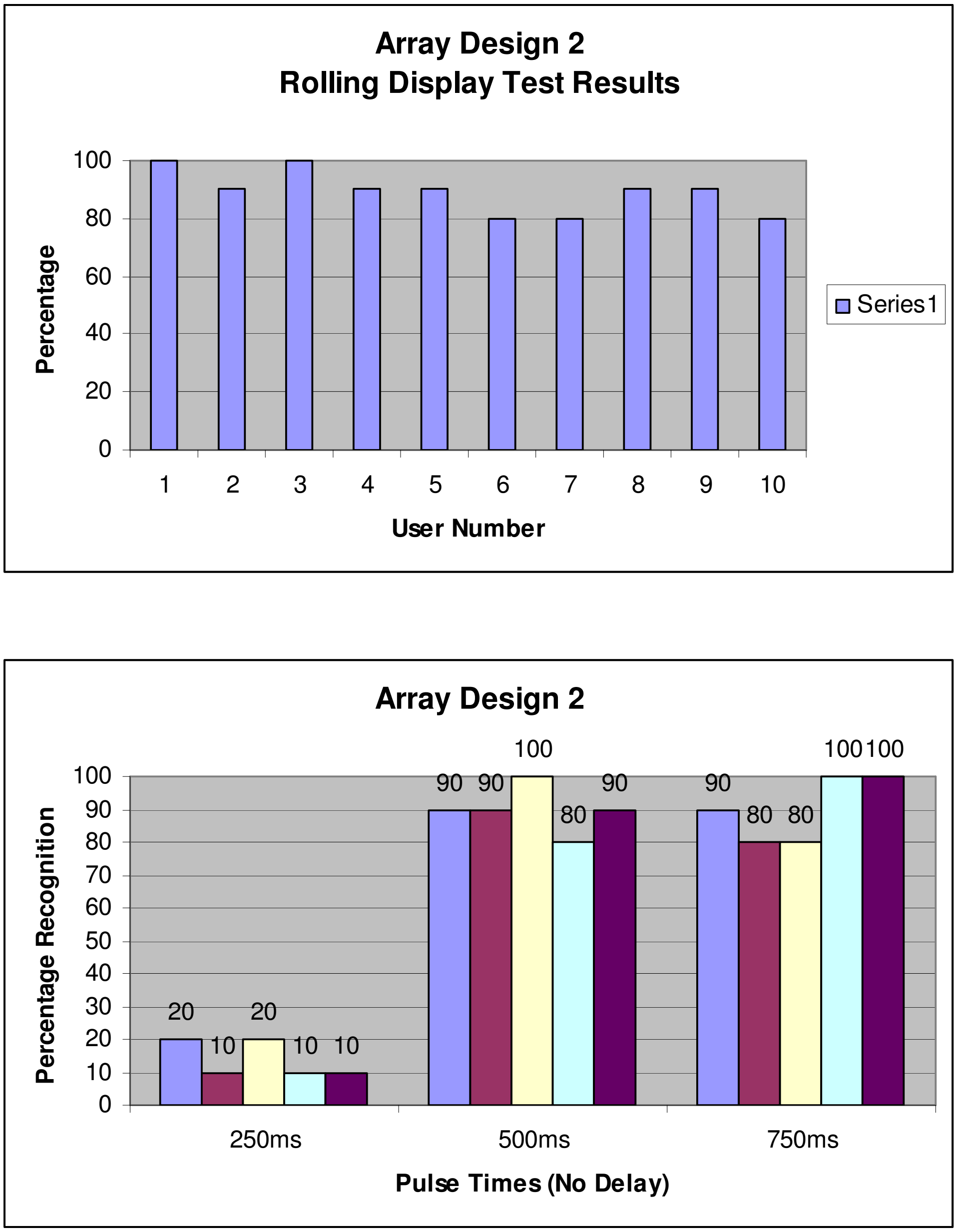


Figure 8.1 Test Results

8.7 Gesture Testing

The testing of the gesture recognition system was integrated into the full system testing and evaluated separately. The actual design of the gestures was outside the scope of the project as the focus was on recognition of hand/arm movements. The mapping of the gestures to their functionality was decided in a semi arbitrary manner, so evaluating the gesture recognition system was limited to asking users to complete a usability questionnaire on using the gestures. Most of the participant (8), had already taken part in my earlier test sessions, while the remaining users had spent at least 15 minutes time playing the game for the test session. This should have been plenty of time to familiarise themselves with the gesture system for them to complete usability questionnaire.

The questionnaire was the System Usability Scale (SUS) [39] as designed by HP was developed as part of the introduction of usability engineering to Digital's integrated office systems programme. Its objectives were to provide an easy test for subjects to complete, to be easy to score, and to allow cross-product comparisons. It has been used extensively in evaluations of projects in Digital and has been found to be simple and reliable. Its goal is to give an overall high level subjective view of usability.

The SUS rates usability on a scale of 0 to 100 and the gesture recognition system rated an average of 87 (over 15 test subjects). This rating showed that the system was usable as an input mechanism; despite the fact the gestures were somewhat arbitrarily designed. The gesture system as an input mechanism for the user test could be said to be effective as user where able to successfully achieve their objectives of directing the player around the maze. Users also reported high satisfaction on questions related to their experience using the gestures while controlling a TV.

8.8 System Testing

To test the viability of using the gesture recognition and a vibrotactile display as a complete input and output interfaces for possible future applications and environments, meant testing the system in an abstract manner. To carry out the testing that I wanted would require a different interaction than controlling electronic devices.

From reading the prior art it should be clear many investigations and tests have been performed using either gesture as input to a system or using vibrotactile feedback as an output mechanism. In chapter 1 I described the goal of the project as investigating new interaction techniques, in particular using gestures as the control/interaction method and using the vibrotactile display as the feedback mechanism. I have not found any research that was conducted using both as the only input and output mechanisms of an interaction with a system with no visual interface.

Using vibrotactile feedback to reduce the cognitive load when other senses are overloaded, or simply as an alternative output modality, has numerous advantages in certain situations. The same statement could also be applied to the use of gesture recognition when compared with other input modalities. To investigate how an integrated inertial gesture and vibrotactile display might be used I have created an abstract scenario for users to complete to demonstrate an interaction. It is hoped that an understanding of the effectiveness of the system can be uncovered and gain an insight into suitable applications and target environments the system might be deployed.

Test 1 – Navigation through a maze using gesture system & vibrotactile feedback.

To test the effectiveness of the overall integrated system I have designed a simple game for users to play. The game involves navigating a person through a maze. There is no visual representation of the maze available to the user as they navigate the person through the maze. This means that the only way to provide feedback to the users is with the vibrotactile display and the only way the user can direct the person around the maze is through hand gestures to indicate direction. The tactile feedback to the user is comprised of three tactons that indicate the direction that the user can turn and one to indicate that the person is walking. This is envisaged as a scenario that people could face in future mediated spaces where the only interaction techniques available to users is gesture for input with vibrotactile feedback.

From these user tests I hope to show how a simple constrained interaction to play a basic game can be easily facilitated by using the integrated gesture recognition system with vibrotactile feedback.

Test 2 – Navigation through a maze while reading.

The objectives of test 1 were to demonstrate an interaction scenario for the gesture recognition & vibrotactile display systems. The objective of test 2 is to test how fast the user completed the test while reading from a book. By forcing the user to read while playing the game should effect the time taken to complete the game, in particular the time take to navigate between intersections of the maps as only similar routes can be compared.

It is hoped that the time taken will not vary widely as the cognitive load placed on the user is to simply remember the direction given through the tactile display and respond appropriately. This should indicate if users are able to perform an interaction with another system while performing a reading task totally unrelated to the game. This will give an indication of the suitability of the system to be used while a user's attention is concentrated on another task.

8.8.1 Test Format

Before the test began the format of the game was described to the users. To encourage them to completed the game in their fastest time they where told that the winning prise was a bottle of wine. The users where given time to practice a test maze to familiarise themselves with the game before completing the test maze using tactile feedback and either audio or visual for the second.

To start the game the users receive the appropriate cue depending on the games feedback mechanism. The first thing the user must do is start the game by gesturing forward. After this they will be given feedback to indicate their options when they arrive at a turn and choose by gesturing left, right or straight. If they arrive at dead end then they receive feedback to indicate dead end. And will be returned to their previous junction where the same options will be given. This way the users can never travel backwards as each junction has a correct and incorrect way to turn. This makes the game as simple as possible and provides the users with an easy task scenario to complete.

8.8.2 Game Restrictions

The game has four intersections where the user must choose left, right or straight ahead. The walking of the player inside maze takes exactly 5 seconds between intersections and if wrong turn is taken it takes 5 seconds to be notified of dead end and 5 seconds to return to intersection where options are displayed again. This format allows me to calculate the time take to complete the games using the one correct route as I can time the correct path through the maze for comparison of times between the two tests. The completion times will be varied by the user's time taken to decide which direction to use. As mentioned earlier user where encouraged to complete the game as quick as possible.

8.9 Analysis of System Testing

The first test was to complete the game to prove the viability of the system and to gain an insight of user reactions to the system and was completed by 15 participants. All users where able to complete the game successfully and guide the player to the end of the maze. To accomplish this users had to be able to understand the feedback that was being provided by the tactile display as to what directions the users could turn. Essentially the users waited for feedback to indicate direction and then gesture in that direction to continue.

7 users did choose an invalid direction; i.e. choosing left when right and straight where the only options, at an intersection and when questioned afterwards all users indicated that they got confused trying to complete the game as quickly as possible. They did not misread what the display was telling them. This is not surprising as when a right turn was available it would be indicated by a vibration on the right hand side of their lower back, or left or top. This is a natural mapping and requires little

cognition to decide which way they can turn, just to gesture in the same direction of perceived vibration of the display.

After users had completed the game they where then asked to play the game again but this time while reading. The results of this test is in no way conclusive but does give an insight into the capabilities of users to perform two completely different tasks simultaneously. The reasons for the doubts of reading too much into the tests results stems from the unquantifiable variables of the test. Example, some users slowed their reading considerable, which I could not quantify, while receiving feedback through the tactile display, while others seemed to not slow in any perceptible manner. This would suggest that some users where either more comfortable using or understanding the interactions, or simply better able to multitask.

Comparing the finishing times achieved while reading and without shows that no user completed the game faster. While this is not unsurprising or unexpected 12 of the 15 users finished within 10% of there first time. Given that the average time taken to traverse the correct path was 43 seconds without reading means that reading while performing the test added less than 4 seconds to their time. One person completed the test within 1 second of there original time while the remaining 4 finished within 20% of their original time.

Again nothing conclusive can be drawn from these tests but they do give an insight into the usability of the system and demonstrates that users could perform another task while playing the game. This will be further discussed in the evaluation.

Chapter 9: Evaluation

At the start of the project I set out to investigate a new interaction technique using gesture recognition and a tactile display. To demonstrate the system it was designed to be used as a control and feedback system for control of electronic devices. Controlling a TV was used in the user testing to define the temporal parameters of the tactile display for displaying tactons to the user. This interaction was limited in what could be discovered about the system with experimentation, so I decided to run a series of simple abstract interactions involving playing a maze game. This was used to show an interaction scenario that utilised my system for input and feedback exclusively. This evaluation will be divided into an evaluation of the individual components and the overall system.

9.1 Gesture Recognition

The gesture recognition was designed to be used as an input mechanism for control or interaction with any suitable system but targeted at controlling electronic equipment for testing and demonstration purposes. At the finish of the project I was able to recognise 8 distinct gestures using one motion sensor attached to the hand on a glove. This number is not the limit of what can be recognised with 1 sensor but gestures become more difficult to differentiate as one data stream limits the scope of what can be detected. This number of gesture provided me with enough input controls for demonstrating the system to control a TV and as input into the maze game.

Again as explained in chapter 2, most of the academic literature on inertial gesture recognition is based on using raw accelerometer data to perform recognition. By choosing to use the Euler angles, provided from the onboard sensor fusion of the MT9, allowed me to avoid having to implement some of the more traditional techniques for recognition such as HMM and Kalman filters. I was able to treat the Euler angles as a temporal data stream on which to perform the recognition. The recognition was based on matching the recorded parameters of performed gesture with the set of recognisable stored gestures, similar to [9].

One problem that must be overcome when using inertial gesture recognition is avoiding the use of a clutch to tell the system that a gesture will be performed. [9] used a button on the input device that the user must press before performing a gesture. This was not an option for me as the targeted scenarios and environments require the cognitive load the system places on the user to be minimal. Therefore a clutch would only add to this confusion. By constraining the gestures and using an activity detection algorithm I was able to perform online gesture recognition.

The overall effectiveness of using the gesture system was elicited using the usability questionnaire, which at 84% would indicate that users where able to use the system in an effective, efficient and satisfactory manner.

9.2 Vibrotactile Array

The tactile array was needed to provide feedback to users without requiring them to use any other modalities and to explore its effectiveness as an output mechanism for the overall system. The testing of the tactile array was carried out in isolation from the gesture recognition to define the parameters that allows tactons to be displayed. After the testing I had gained the necessary parameters need to display the tactons and had gained experience in using the array to provide tactile feedback.

Very early on in testing it became clear that a constant display pattern was not a suitable method with which to display patterns due to user being unable to distinguish any pattern. This is supported by [20] who states that placement of vibration tactors in too close proximity can cause the senses being unable to distinguish between them.

During the test sessions for the tactile array it also became apparent that users could not perceive the exact patterns that where being displayed. By this I mean the geometric shape of the pattern. For instance when displaying the LeftToRight display pattern users responded by saying that they could perceive movement across their back from left to right. The same sensation was described for the opposite display pattern as well as the top down and anti clockwise patterns.

Although I was not aware of this at the time but the sensations described where actually what is known as the sensory saltation phenomenon [25]. This is the process where the sensation that is perceived by a person is different that the actual stimulation pattern. This would explain why users where reporting the sensation of something moving across their back instead of perceiving 4 localised stimulation pulses for each tactor row that was activated.

I would like to have performed alot more testing with the tactile array to examine the full range of tactons that could be distinctly perceived by users with the 4x4 tactile array. I would hypothesise that the tactile display would reach a limit pretty quickly on the amount of tactons that could possibly be distinguished by the user, which would give an indication as to the range of messages that could be displayed cutaenously to users. This would help future designers who are thinking of using the array as a feedback mechanism to decide the effectiveness of using a tactile display of a similar design.

Having the ability to control the intensity of the stimulation exerted by the tactors would also have given me the ability to experiment more with the tactons. For instance by varying the intensity at the start or end of a pattern gives you the ability to add some rhythmic meaning to the display. This was shown to very good effective for [14] who tested tactile messages using one tactor on the end of a finger to tell a user if they where receiving a text, MM or voice message by varying the rhythm of the stimulus.

9.3 Integrated Gesture Recognition and Tactile Display

The testing for the overall system was deliberately abstract. I felt that using the TV scenario that the system was originally designed to operate with would not allow me explore the overall system in the manner that I wanted. Although the system performed excellently when used as the remote control system for TV and could be extended to operate as a generic controller of such devices. I could not devise a test session that would allow me to test the systems ability to operate as an independent interface to a system as the TV would provide direct feedback, rendering the feedback to users from the tactile display redundant in term of whether the command was accepted.

For the first test session I disguised a constrained interaction in the form of a maze game. This made the test session a lot more enjoyable for users and encouraged them to complete the interaction as fast as possible, although there where other enticements. The test showed the viability of the system to operate as the standalone interface to the game. To accomplish this meant the user had to have the meaning of the tactile feedback messages explained and some practice to familiarise themselves with them. After this initial period users became completely comfortable with the interaction mechanism and played the game quite easily.

This training period would be needed with most interactions of this type as any new interaction using the gesture and tactile mechanism would require the user to learn what the system was communicating to them through the tactile feedback. The intended communication into the system would also have to be mapped to the gestures, which users would also have to learn. It is the freedom of choosing these variables into and out of the system that open up the possibility of new interactions being created.

The second test was carried out to access the effect of playing the game would have while someone was preoccupied with reading. This test was needed to test the viability of using the gesture and tactile mechanism of communication to operate when the user's attention was focused on another task. This was hoped to mimic possible interaction scenarios, as discussed in the introduction and prior art chapters, where users are occupying public mediated space within a building but wish to continue interacting with a computer system. Or in situations where the other modalities are occupied leaving only gesture and tactile feedback for communication.

The results certainly hint that this is a feasible alternative with some users not suffering any perceptible delay in completing the game while having to read. Again there are too many variables, such as user ability to multitask and attention paid to reading material to draw any definitive conclusions. This also gives us an insight into the different environments where the system might be deployed and where it would be most effective. Any environment where distracting the users attention with visual or audio communication is not safe or feasible would suit the system as communication to the user through tactile feedback can be accomplished without detracting from the user current task. Examples could be drivers of cars or aircraft pilots whose visual attention would be near saturation point already. People who operate in extremely noisey environment and have to wear ear protection could have information communicated cutaenously.

After completing both tests I thought about extending the concept of the maze game. What if I didn't constrain the interaction to return the player to intersection where they took the wrong turn and lead them down a false trail instead? Their chances of finding the correct path to the finish increase exponentially with each intersection. Using a number combination of four digits, similar to a typical pin number but instead of representing a code are the directions through a maze, i.e. 9th left 1st right 4th right 7th left. The probabilities of someone choosing the proper combination are high enough to prevent someone guessing. So why not use the system as a lock and key mechanism for access to a safe. Certainly there are drawbacks such as not allowing someone observe the interaction, but this is the case with all pin numbers that must be entered via a keypad. It might take some time to complete the interaction at the size stated above with 4 options of 9 turns but this could be scaled to suit the security risk.

Maybe this type of scenario of challenge response interactions would be the most suitable for the system to operate within. Extending the range of this interaction would involve training the users for an appropriate period to become familiar with the implemented communication system. The more complicated the system needs to be, by extending number of input gestures and display tactons, would correlate to amount of training needed to master and become comfortable with system.

Chapter 10: Conclusions

In this chapter the entire project is considered. I will begin with a summary of the project and then comment on future improvements. Overall the work of my dissertation has demonstrated a novel interaction technique when integrated into a game and have used it as controller for a TV. I have also shown that the system could be suitable to certain environments/situations, where information can be communicated to the user through tactile feedback and back through gesture, while they are focused on another task. Further I have stated how these interactions could be extended to act as a challenge /response mechanism to operate as an entry code to demonstrate other interactions scenarios that are possible or suitable for the integrated gesture recognition and tactile feedback system.

10.1 Summary

The integrated gesture recognition and feedback system uses an MT9 motion sensor that is equipped with an accelerometer and gyroscope to capture the data that is generated by users performing hand/arm gestures. The onboard sensor fusion allows Euler angles to be used as output from the sensor. This data is first analysed by an activity detection algorithm to determine the start of a gesture, which is then tracked and recorded. The recorded gestures parameters are matched in a quasi fuzzy logic type manner to a set of recorded gestures or determined to be invalid. A total of 8 distinct hand arm gestures where recognisable by the system using one motion sensor.

The vibrotactile array was constructed using a 4x4 formation of coin type vibration units. The array was controlled from the Keil MCBx51 development board, which in turn could be controlled using an external RS232 serial connection. During testing the array was able to display 8 different tactons to users that where recognisable. I was also able to recreate the sensory saltation phenomenon when displaying the tactons to the users, although not aware of it at the time.

Through user testing I have shown that the system is usable as the only control/feedback mechanism of an interaction by using the system to play a game. The simple constrained maze game showed an example of what could be accomplished with the system and in my evaluation I mention about extending the concept to operate as an access control mechanism. The overall interaction here is based on a challenge response type paradigm, which maybe hints at the suitable scenarios applicable to the integrated system.

10.2 Future Work

As already stated in the evaluation I would like to carry out more testing to more fully access the capabilities of the integrated system. This could also mean redesigning the array to make it more suitable for this type of experimentation by having the ability to vary the intensity of the stimulation when using the tactile display. The gesture recognition could also be extended by placing more sensors on the arm and even onto both arms. Both these extensions would allow more complex interactions to be investigated.

I think the main area for future expansion of the work would be investigating integrating the system into an overall multimodal interface. This would raise many questions into when to use the extra modalities provided? Also questions of how combining the modalities simultaneously would affect users of such a system.

As the integrated system can be tailored to provide “private” interactions in the sense that no other person can perceive from observing the person using the system just what they are doing. These private interactions could also be investigated with possibly also looking at what communication could be possible with two humans communicating.

Bibliography

[1] Weiser, M. 1995. The computer for the 21st century. In *Human-Computer interaction: Toward the Year 2000*, R. M. Baecker, J. Grudin, W. A. Buxton, and S. Greenberg, Eds. Morgan Kaufmann Publishers, San Francisco, CA, 933-940.

[2] Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P.: Project aura: Toward distraction-free pervasive computing. IEEE Pervasive computing (2002) 22-31 4 http://citeseer.ist.psu.edu/garlan02project.html

[3] Meyer, S. and Rakotonirainy, A. 2003. A survey of research on context-aware homes. In *Proceedings of the Australasian information Security Workshop Conference on ACSW Frontiers 2003 - Volume 21* (Adelaide, Australia). C. Johnson, P. Montague, and C. Steketee, Eds. Conferences in Research and Practice in Information Technology Series, vol. 34. Australian Computer Society, Darlinghurst, Australia, 159-168.

[4] Rudolph, L. 2001. Project Oxygen: Pervasive, Human-Centric Computing - An Initial Experience. In *Proceedings of the 13th international Conference on Advanced information Systems Engineering* (June 04 - 08, 2001). K. R. Dittrich, A. Geppert, and M. C. Norrie, Eds. Lecture Notes In Computer Science, vol. 2068. Springer-Verlag, London, 1-12. [5] Home lab – Phillips

[5] IBM systems journal vol 38, no 4, 1999 W. Mark.

[6] Benbasat, A. Y. and Paradiso, J. A. 2001. Compact, configurable inertial gesture recognition. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (Seattle, Washington, March 31 - April 05, 2001). CHI '01. ACM Press, New York, NY, 183-184. DOI= http://doi.acm.org/10.1145/634067.634178

[7] Welch, G. and Bishop, G. 1997. SCAAT: incremental tracking with incomplete information. In *Proceedings of the 24th Annual Conference on Computer Graphics and interactive Techniques* International Conference on Computer Graphics and

Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., New York, NY, 333-344. DOI= http://doi.acm.org/10.1145/258734.258876

[8] Ko, T., Demirdjian, D., and Darrell, T. 2003. Untethered gesture acquisition and recognition for a multimodal conversational system. In *Proceedings of the 5th international Conference on Multimodal interfaces* (Vancouver, British Columbia, Canada, November 05 - 07, 2003). ICMI '03. ACM Press, New York, NY, 147-150. DOI= http://doi.acm.org/10.1145/958432.958461

[9] Lementec J-C and P.Bajcsy, “Recognition of Arm Gestures Using Multiple Orientation Sensors: Gesture Classification,” the 7th International IEEE Conference on Intelligent Transportation Systems, Washington, D.C., October 3-6, 2004

[10] Brewster, S., Lumsden, J., Bell, M., Hall, M., and Tasker, S. 2003. Multimodal 'eyes-free' interaction techniques for wearable devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM Press, New York, NY, 473-480. DOI= http://doi.acm.org/10.1145/642611.642694

[11] Brewster S, “Multimodal Interaction and Proactive Computing”, http://www.dcs.gla.ac.uk/~stephen/papers/proactive_computing_paper.pdf

[12] Brewster, S. 2002. Overcoming the Lack of Screen Space on Mobile Computers. *Personal Ubiquitous Comput.* 6, 3 (Jan. 2002), 188-205. DOI= http://dx.doi.org/10.1007/s007790200019

[13] Brewster,S.A. and Brown, LM., Tactons: Structured Tactile Messages for Non-Visual Information Disply. In Proceeding of Austraillian User Interface Conference 2004, Austraillian Computer Society, 15-23.

[14] Brown, L., Brewster, S.A. and purchase, H., A first investigation into the effectiveness of Tactons. In Proceedings of World Haptics 2005, (Pisa, Italy, 2005), IEEE Press.

[15] Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. 2000. Sensing techniques for mobile interaction. In *Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology* (San Diego, California, United States, November 06 - 08, 2000). UIST '00. ACM Press, New York, NY, 91-100. DOI= http://doi.acm.org/10.1145/354401.354417

[16] Perera, S. (2002). Tactile Perception and Design http://www.tiresias.org/reports/tpd1.htm A report that forms part of the DTI funded Bionic Project.

[17] Lechelt, E.C. (1984). Tactile Temporal Acuities. Annals of the New York Academy of Sciences. Vol 432. pp 612 - 613.

[18] Kontarinis, D.A. & Howe, R.D. (1995). Tactile Display of Vibratory Information in Teleoperation and Virtual Environments. Presence. Vol 4. No 4. pp 387 - 402.

[19] Pawluk, D.T.V., van Buskirk, C.P., Killebrew, J.H., Hsiao, S.S. & Johnson, K.O. (1998). Control and Pattern Specification for a High Density Tactile Display. Haptics Symposium. USA.

[20] Raj, A., Roetzer, L., Fatolitis, P., Cholewiak, R., Kass, S., & Rupert, A. (2001). Haptic Interfaces to Augment Human Machine Interactions in Space Activities.

[21] Geldard, F.A. (1957): Adventure in Tactile Literacy. The American Physcologist, 12, 115-124.

[22] Tan, H.z and Pentland, A., “Tactual displays for sensory substation and wearable computers.” Fundamentals of wearable computers and augmented reality, Lawrence Erlbaum Associates: Mahwah, New Jersey, 2001. pp 597-598

[23] Craig, J.C. and Sherrick, C.E., “Dynamic Tactile Displays”, Tactual Perception: A sourcebook, W. Schiff and E. Foulke, eds, Cambridge University Press, 1982. pp209-233.

[24] Tan, H. Z. and Pentland, A. 1997. Tactual Displays for Wearable Computing. In *Proceedings of the 1st IEEE international Symposium on Wearable Computers* (October 13 - 14, 1997). ISWC. IEEE Computer Society, Washington, DC, 84.

[25] F.A. Gerarld (Ed), Cutaenous Communication Systems and Devcices, The Psychonomic Society, Inc, 1973.

[26] Lindeman, R.W., Cutler, J.R. "Controller Design for a Wearable, Near Field Haptic Display." Haptic Interfaces, 2003, Haptic 2003, Proceeding, 11th Symposium, 22 March, 2003, pp 397-403

[27] XSens Technologies. Motion Tracker technical documentation. http://xsens.com

[28] Hewlett-Packard. HP IPaq 5500 Series product overview. http://h200002.www2.hp.com/bc/docs/support/SupportManual/c00046402/c00046402.pdf, 2003

[29] Total Remote Audio IR Dongle, Griffin Technologies. User Manual http://www.griffinserver.com/downloads/pdf/manuals/Total_Remote_user_manual.pdf

[30] Keil Embedded Development Tools. http://keil.com/mcbx51/

[31] Chip Catalogue – Free Datasheets www.chipcatalog.com/TI/ULN200AN.htm

[32] TechGSM. http://www.techgsm.com/cat_Vibration_motor_347.html

[33] Socket Communications Inc. Ruggedised dual serial I/O pc card. http://www.socketcom.com/pdf/io/diopcdatasheet.pdf, 2003

[34] Brainboxes, Serial Card Solutions http://www.brainboxes.com/product/pcmcia/PM-132.aspx

[35] 8051 Tutorial. http://www.8052.com

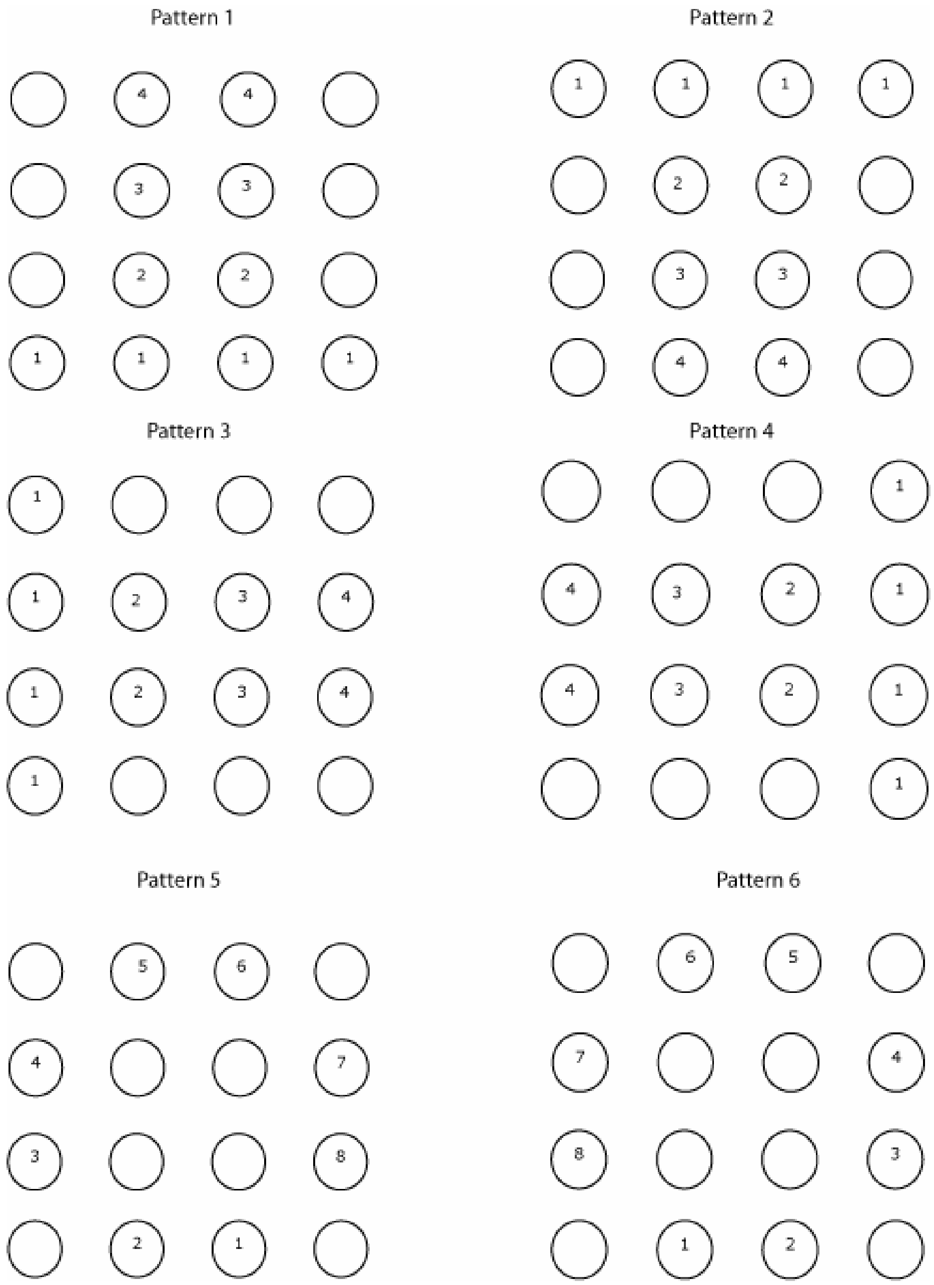
[36] Jinlong Machinery. http://www.vibratormotor.com/cyl-table.htm

[37] Bill Pugh – Component Object Model Overview. http://www.cs.umd.edu/pugh/com ,1998

[38] IRDA Explanation. http://www.mobilecomms-technology.com/projects/irda/

[39] Simple Usability Scale. White Paper by John Brooke at, www.usabilitynet.org/trump/documents/Suschap.doc

Appendix A – Patterns Displayed by Tactile Array



Pattern 1

	4	4	
	3	3	
	2	2	
1	1	1	1

Pattern 2

1	1	1	1
	2	2	
	3	3	
	4	4	

Pattern 3

1			
1	2	3	4
1	2	3	4
1			

Pattern 4

			1
4	3	2	1
4	3	2	1
			1

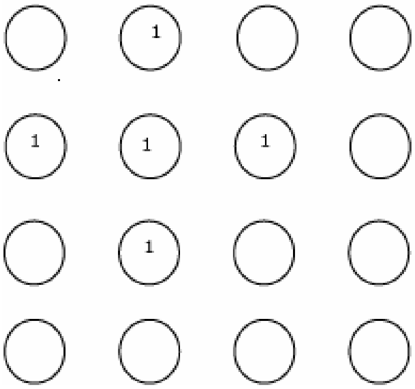
Pattern 5

	5	6	
4			7
3			8
	2	1	

Pattern 6

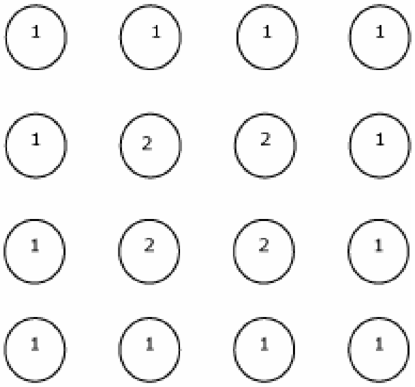
	6	5	
7			4
8			3
	1	2	

Pattern 7



	1		
1	1	1	
	1		

Pattern 8



1	1	1	1
1	2	2	1
1	2	2	1
1	1	1	1