Fast Multi-Axis Tracking of Magnetically-Resonant Passive Tags: Methods and Applications

By

Kai-yuh Hsiao

B.S., Electrical Engineering and Computer Science (1999)

Massachusetts Institute of Technology

Submitted to the Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Computer Science and Electrical Engineering

At the

Massachusetts Institute of Technology

February 2001

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
February 1, 2001

Certified by . . . .
Joseph A. Paradiso
Principal Research Scientist, Media Arts and Sciences
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Fast Multi-Axis Tracking of Magnetically-Resonant Passive Tags: Methods and Applications

By

Kai-yuh Hsiao

Submitted to the Department of Electrical Engineering and Computer Science
On February 6, 2001 in partial fulfillment of the requirements for the
Degree of Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

We have explored the design and development of magnetically-resonant tag readers for application to tangible computer interfaces. To this end, we constructed a ringdown tag reader and a swept-frequency tag reader, both capable of real-time continuous interaction with multiple tagged objects. Although the ringdown reader worked well for smaller numbers of tags, the swept-frequency tag reader proved more efficient for work with twenty to thirty tags, and so we did further work to extend and apply it.

Graphical and musical applications were developed for the swept-frequency tag reader and proved its usability for driving tangible desktop interfaces. Finally, a six-coil variant was constructed in order to determine the three-dimensional position and orientation of tagged objects. This thesis describes both reader systems, outlines our demonstration applications, and gives first test results from the multiple-coil tracker.

Thesis Supervisor: Joseph A. Paradiso

Title: Principal Research Scientist, Media Arts and Sciences

Acknowledgements

First and foremost, I would like to thank Joe Paradiso, who is my advisor, supervisor, colleague, and friend, and whose analog circuit wizardry provided the vast majority of the basic concepts and the underlying electronics for my work on these projects. Joe has given me nothing less than superhuman amounts of his time, attention, and patience, and his singularly amazing spirit, supportiveness, and engineering abilities are an inspiration to us all. Furthermore, I would like to thank the guys at British Telecom and the rest of the Things That Think consortium for funding the past 1.5 years of this endeavor, and also my parents for funding the several years preceding that. Thanks also to the members of the ever-growing Responsive Environments group, especially Ari Benbasat, whose occasional help and advice, along with his seemingly unwitting support role for our entire group, often provided the extra kick necessary to just get things working. And finally, thanks to the MIT Media Lab, and everyone who helps it to run smoothly, for making every part of this experience even possible.

About the Author

Kai-yuh Hsiao's primary interests in computers and music started at an early age. He has been working at the Media Lab since his arrival at MIT as an undergraduate in 1995, and all of his projects have involved some combination of computers, electronics, and music. After receiving his bachelor's degree in 1999, he continued working under Joe Paradiso in the Responsive Environments group in order to finish his master's degree.

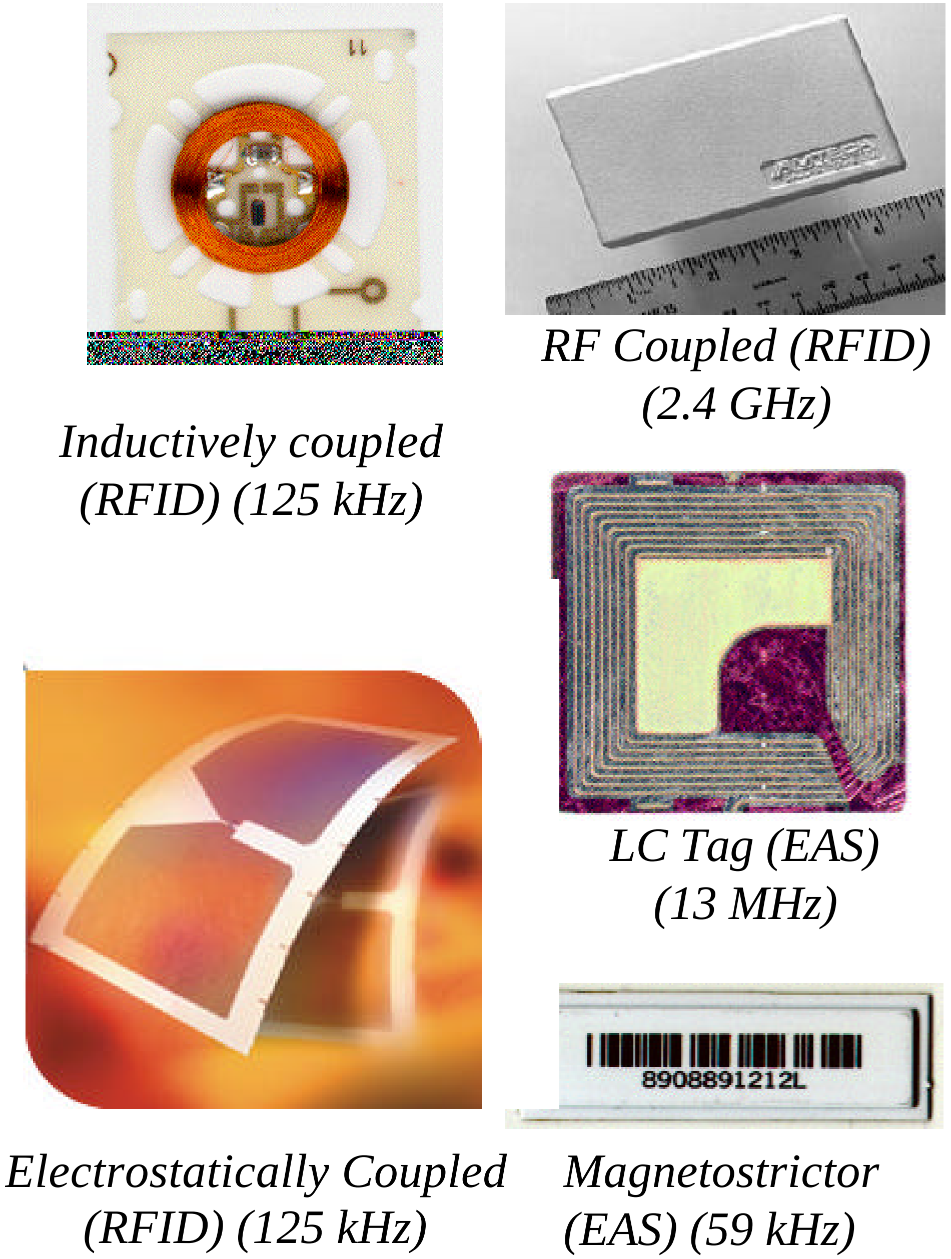
Table of Contents

Chapter 1

Introduction

An important direction that computer-human interface research has taken in recent years focuses on enabling any object, anywhere, to interact with a computer. For instance, the "tangible interface,"[1] where the manipulation of physical, so-called “tangible” objects is observed by a computer and used as input, is a topic of much exploration, in the hopes that someday soon, interacting with a computer will be as natural as manipulating standard, everyday items. The present-day graphical user interface (GUI) paradigm uses keyboards, mice, and 2D displays as interface devices, but despite its versatility, the user is still fairly remote from the data being manipulated. This stiff and somewhat artificial mode of interaction will gradually be augmented by a physical dialog with real world objects as they become interfaces to an omnipresent ubiquitous computing environment.[2] Being able to control a computer by simply manipulating objects on a physical desktop would allow a seamless, intuitive interface well beyond what is commonly possible today on the virtual desktop.

One approach to producing tangible objects is to embed each object with a small wireless tag.[3] These tags exhibit electromagnetic behavior that enables a suitable tag reader device to detect their key properties, such as location and identification, remotely. With a tag attached to the relevant objects, computers can observe their movements and manipulations and use these data as input. In this paper, we describe the development of several wireless tag readers well suited to these concepts and introduce several applications of them as tangible interfaces.



Inductively coupled (RFID) (125 kHz)

RF Coupled (RFID) (2.4 GHz)

LC Tag (EAS) (13 MHz)

Electrostatically Coupled (RFID) (125 kHz)

Magnetostrictor (EAS) (59 kHz)

Figure 1. Various types of common tags.

1.1. Types of wireless tags

The term "wireless tag" describes any of a range of small, inexpensive electronic circuits with electromagnetic properties that enable a tag reader system to detect the tags and possibly read other properties of the tags as well. Several examples of wireless tags are shown in Figure 1 and are described below. Wireless tagging is already a large, well-established industry, and some wireless tagging technologies are already in very common usage. Perhaps most popular are the ubiquitous “Electronic Article Surveillance” (EAS) tagging systems found in many shops today; in order to prevent shoplifting, small wireless tags are embedded on or in merchandise packaging of the items stocked in a store. When the tags come within range of a reader device placed at the exit to the store, they are detected, alarms ring and the shoplifter has been caught. When the item is purchased properly, the tags are deactivated so they no longer respond and the item can be removed without incident.

As prevalent as they are, the common types of EAS tags are a mere subset of the types of tags that have been developed for a multitude of applications. First, tags fall into two primary categories, active and passive. Active tags draw upon a source of power (e.g. battery) attached directly to the tag. These tags typically operate in a low-power mode, waiting for an appropriate signal to wake them up. Whenever they receive such a signal, they draw upon their power sources to transmit their programmed information. Passive tags, in contrast, have no attached power source, and instead they use the power transmitted by the detection antenna.

Within the category of passive tags are RFID tags, as well as a whole assortment of EAS tag types. RFID tags have an onboard microchip, which upon receiving power either sends a signal directly back or manipulates the field from which they receive power, for example by modulating the impedance of their antennae. This enables their stored information to be read and/or written. EAS tags, on the other hand, are typically less complex, contain no integrated circuits, and manipulate the reader’s fields in a simple and detectable fashion. In this thesis we will concern ourselves with several types of EAS-style tags, which are in turn a subcategory of passive tags.

RFID tags can be split into categories based on the frequency and type of field they operate in. High-frequency tags derive their power from a rectified RF antenna (“rectenna”) and commonly operate around 2.4 GHz. These tags typically transmit data back to the reader via RF reflections of the interrogating signal, e.g. by modulating their antenna impedance. Magnetically-coupled tags operate at lower frequencies (typically at 125 kHz or 13 MHz). Transmit coils at the reader produce a magnetic field which is received by a coil wound around the chip. Power is inductively coupled into the chip tag's coil (analogously to a transformer), which is also used to transmit and receive data to and from the reader. Finally, electrostatic tags[4] operate on a similar principle, but instead of a coil they use two flat metallic plates, which extract power and communicate capacitively via a changing electric field.

The other general class of passive tag, which the remainder of this thesis will be working with, is the resonant EAS tag. This class of tags is very appropriate for simple detection applications, most notably shoplifting detection. Compared to the more complex RFID tags, these tags are often known as "1-bit tags", because they traditionally provide only one item of information: whether they are present in the detect field or not. The tags used in these applications tend to be simple and cheap, and so they produce only simple responses, thus providing a limited space of ID codes, as opposed to the more complex dialog possible with chip-based RFID tags, which can store arbitrarily long ID bitstrings.

Currently, the most commonly found type of shoplifting tag is the magnetostrictor.[5] This extremely inexpensive tag is simply a magnetically-coupled strip of metallic glass with mechanical properties that create a high-Q resonance (meaning their resonance response is very strong and responds to very specific frequencies) in the sub-100 KHz range, so when it receives a magnetic pulse at its resonant frequency, it produces a "ringdown" response which can be detected by the reader. Because the response of the tag is dependent on a small strip magnet attached to the tag, these tags can be deactivated by demagnetizing the strip magnet. This makes it convenient for stores to turn off these tags by bringing the tags near an erase coil, allowing customers to leave with their purchases without incident.

Another simple such resonant tag is the LC tag, which consists of an inductor (L) in series with a capacitor (C). This, too, is magnetically coupled, through its inductor component, and it is electromagnetically resonant, making it similarly detectable. This type is usually cut out of a flat sheet of conductive foil and can thus be used for shoplifting detection for items such as books or boxes.

Two other examples of shoplifting tags are the magnetic wire tag,[6] which uses a length of nonlinear magnetic wire to produce detectable high-order harmonics when subjected to an AC magnetic field; and the microwave diode tag,[7] which uses a diode whose bias varies when exposed to a capacitively-coupled 100 kHz electric field generated near the store's exit, thus producing detectable harmonics that distort an ambient microwave signal reflected by the tag.

1.2 Magnetically resonant tags as sensors

The magnetostrictor and LC resonant tags have similar electromagnetic properties. They both resonate and couple magnetically at relatively low frequencies, and thus both types can be detected by similar tag readers. An appropriately designed tag reader would be able to detect the presence and proximity of such tags to a coil generating an appropriate AC magnetic field.

Typically these tags are used to provide their single bit of ID, i.e., whether they are present or not. However, there have been several projects that examine more complex sensing capabilities of such tags.[8,9] These center around adjusting physically variable parameters of the tags and detecting such changes using specialized readers. Specifically, tags could be constructed such that their resonant frequencies or Q vary in proportion to some other parameter, such as temperature, pressure, or local DC magnetic field.

Once such a shift in tag properties can be detected, these tags then enable yet another dimension of manipulation aside from simple presence and position. For instance, an LC tag could be embedded in the bottom of a milk carton, with an elastic material between the capacitor plates.[8] The presence and position of the milk carton can then be identified, by virtue of the distinct resonant frequency of the attached tag and the magnitude of its returned signal. Additionally, the quantity of milk remaining in the carton could furthermore be measured, because the capacitor’s plate spacing, which determines the capacitance and thus the tag’s resonant frequency, is shifted as a function of the weight of the remaining milk.

1.3 Tag readers for real-time CHI interaction

Our purpose in working with tagging techniques was to design and explore systems that could enhance real-time interaction with a computer, particularly to enable and experiment with useful tangible interfaces. The systems available commonly on the market tend to focus either only on presence detection of a small set of distinct tags over a reasonably large range, which is the case with shoplifting-detection systems, or only on detection of a large number of tags somewhat slowly and not so cheaply, which is the case with chip tags, which have to use expensively manufactured onboard chips with time-consuming anticollision protocols designed to prevent multiple tags from interfering with each other during operation. Essentially no chip tags are optimized specifically for real-time, continuous tabletop interaction between a user, an array of several distinct physical objects, and a computer.

For the purposes of this paper, we have explored the use of magnetically resonant (magnetostrictor and LC) tags in conjunction with two tag reading principles, pulsed ringdown and continuous swept frequency, in order to develop an efficient and inexpensive tag reader easily applicable to a wide variety of tangible interfaces.

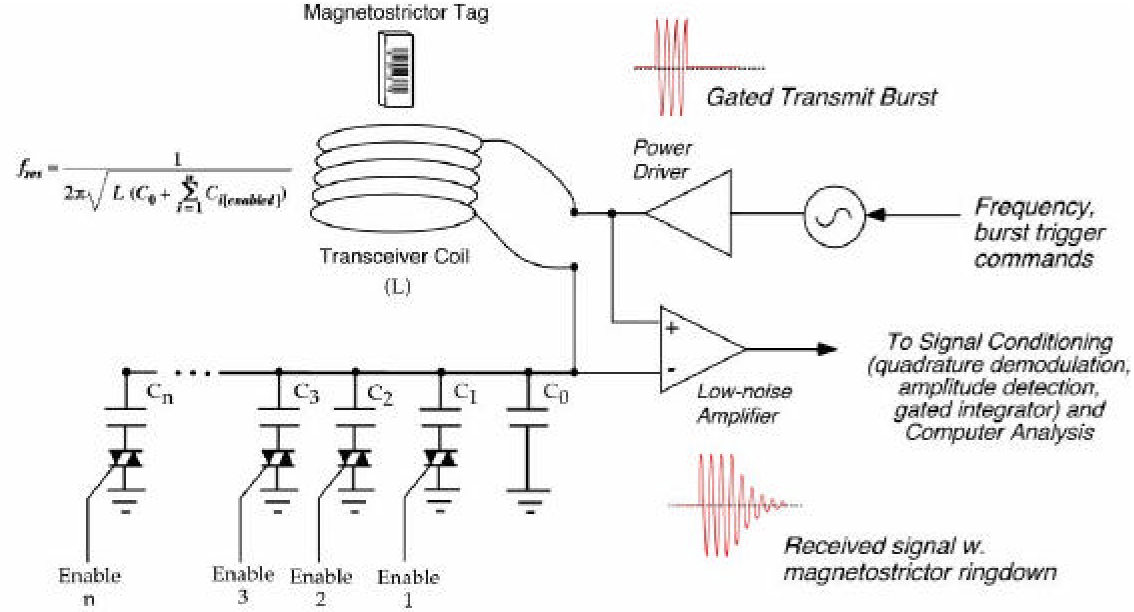
The next two chapters introduce the two families of tag readers that we have developed and the following two chapters present example applications and extensions.

Chapter 2

Ringdown tag readers

The first family of tag readers we examined was the "ringdown" tag reader. This type of magnetically resonant tag reader works on a principle similar to that used in pulse-induction metal detectors[10] and nuclear magnetic resonance imaging systems.[11] This family of tag readers, already in fairly common usage in EAS systems, takes advantage of the resonance properties of high-Q magnetostrictor and LC tags in order to detect them.

The basic operation of such tag readers can be described in three steps, as diagrammed in Figure 2. First, a magnetic pulse is transmitted by the reader at the resonant frequency of the desired tag. Next, the tag, upon exposure to this pulse, produces



Magnetostrictor Tag
Gated Transmit Burst
$f_{res} = \frac{1}{2\pi\sqrt{L\,(C_0 + \sum_{i=1}^{n} C_{i[enabled]})}}$
Power Driver
Frequency, burst trigger commands
Transceiver Coil (L)
+
-
To Signal Conditioning (quadrature demodulation, amplitude detection, gated integrator) and Computer Analysis
Low-noise Amplifier
C_n
C_3
C_2
C_1
C_0
Received signal w. magnetostrictor ringdown
Enable n
Enable 3
Enable 2
Enable 1

Figure 2. Basic block diagram of the ringdown tag reader.

its own mechanical or electronic resonant response, which will continue for a short while as the tag damps out after the transmitted pulse stops.   Finally, because the tag produces its own weak resonant signal, the magnetic drive at the reader can be turned off once the tag rings up, and then the reader "listens" for this small but detectable response.

The response of the tag, which does not cut off immediately but rather decreases exponentially over time, is described as a "ringdown," and because these systems listen for this ringdown response, they are called ringdown tag readers. Ringdown tag readers have a number of advantages. First, they have a good range; the transmit pulse can be very powerful and so can easily stimulate tags across a range of several meters. For example, common EAS systems can detect tags up to three meters away.[12] Furthermore, the configuration can be fairly flexible; one transmit antenna can broadcast the signal, and then either the same coil can be used for detection, or other antennas can also be used to listen for the response from various locations. Also, the design of these circuits is fairly straightforward. Finally, the time required to detect one tag can be made relatively short, with ring-up times within a millisecond or so, compared to the more complex chip-based RFID systems, which typically require more time to power up CMOS chips and transmit back ID's. These advantages led us to design and examine ringdown tag readers. Enabling a ringdown reader to have continuous real-time interaction with multiple tags was the first step.

2.1 Design of a continuous real-time ringdown tag reader

The ringdown reader has two responsibilities; it must transmit a magnetic pulse ("ping"), and then it must stop that pulse and listen for a response from a tag. However, in order to satisfy our requirements for a tangible interface backend, several additional specifications must be added to the list.

First, we would like to detect multiple tags. For each tag that we wish to detect, this requires that we transmit a magnetic pulse at the tag's resonant frequency. Second, we must design the receive circuitry to detect a ringdown signal at the frequency of each of the tags. Furthermore, in order to make the system continuous and real-time, the reader must be able to distinguish the magnitude of the ringdown signal, and it must be able to perform the transmit-receive cycle fairly quickly.

2.2. Implementation of ringdown reader transmit circuitry

The first goal was to be able to transmit a pulse for the detection of multiple tags at various resonant frequencies. Two approaches were considered for this objective. First, the transmit antenna could simply be wired to transmit a burst of broadband noise, which has components at all frequencies of interest. This noise would stimulate all tags within range and provoke ringdown responses from all the tags at once. Second, the transmit antenna could cycle through a known list of resonant frequencies, pinging briefly at each one and listening for a response.

Despite the relative simplicity of the broadband noise approach, we opted not to use it because the transmission would require a large amount of power in order to provide sufficient excitation at all relevant frequencies. Furthermore, the receiver would then be somewhat complex; the simultaneous ringdown response of all the tags at once would create the need for a real-time Fourier transform in order to determine which tags were responding at the relevant frequencies.

Once it was determined that pinging sequentially on multiple known frequencies would be more straightforward to implement, the issue became finding an efficient means with which to do so. Generating and transmitting a relatively powerful signal at a given set of frequencies was not prohibitively difficult. As seen in the schematics provided in the appendix, we used a voltage-controlled oscillator to provide the signal, and the amplification of the signal was done using a pair of power MOSFET's that switched the drive coils across the 40-volt supply rails.

One of the difficult design issues was finding a way to drive a similarly large amount of current through the coil setup at all the relevant frequencies, in order to produce a reasonably large magnetic field. Because the transmit coil would have its own resonant curve, frequencies further from the coil's resonance would tend to produce weaker transmitted signals. The apparent solution, then, was to change the resonant frequency of the coil dynamically, tuning it to match the frequency of the ping being transmitted.

The resonant frequency of the coil assembly, like all inductor-capacitor resonant circuits, would be given by $f = \frac{1}{2p\sqrt{LC}}$, where L is the inductance of the coil and C is the series capacitance. The coil itself, which represented the inductive load of the resonance, could not be changed, and thus the coil we used had a constant inductance around 120 uH. However, the capacitors connected to the coil could be electrically switched in and out, and this would enable real-time adjustment of the coil's resonance. The two possibilities were 1) to switch continuously and quickly between two capacitors, allowing each one to have a partial impact on the total resonance of the system, as in a switched-capacitor filter,[13] or 2) to find some way to switch discretely between several capacitors. The first approach would have the advantage of allowing any resonant frequency to be attained between the values of the two attached capacitors, by adjusting the switching duty cycle. However, it proved to be inefficient because of losses in signal due to the continuous, high-voltage, high-frequency switching in the power MOSFET's that we used. This resulted in a low Q, meaning that the coil ended up transmitting too inefficiently.

As a result, we implemented the second approach instead, using a binary-weighted ladder of six sensitive-gate triacs and capacitors, which enabled us to switch the high-power AC signals necessary for transmission using simple logic level signals. As seen in the schematic given in Appendix 1, each triac was connected to capacitors of different values, and thus the capacitance of the coil circuitry could be switched between various values, enabling a discrete shift in resonance for pinging more efficiently, and thus more powerfully, at different frequencies. Thus, when generating pings, the transmit system first activated the proper triacs, and hence switched the proper capacitors into place, and then the drive signal at the respective frequency was sent to the amplifier and coil. Each ping was generated for about 1.5 milliseconds, and then the transmit circuit was rapidly damped in order to silence it and listen for a ringdown response. A circuit board with this transmit circuit, including the MOSFET drivers, the front-end receive amplifiers, the triacs and all the capacitors, can be seen in Figure 3.

Having selected a transmit circuit design that was optimized to ping efficiently at a set of discrete frequencies, we then had to select a set of tags that would respond well and be reasonably optimized for detection by a receive circuit with our specifications.

2.3. Selection of tags

Once the transmit circuitry was designed, our next step was to decide on a type of tag to use for detection. The choice was between magnetostrictor tags and LC tags, which have differences in response power, resonance strength, and ease of tuning.

Magnetostrictor tags have a higher Q value than LC tags. This means that in general magnetostrictors have lower losses and a sharper frequency response, hence they

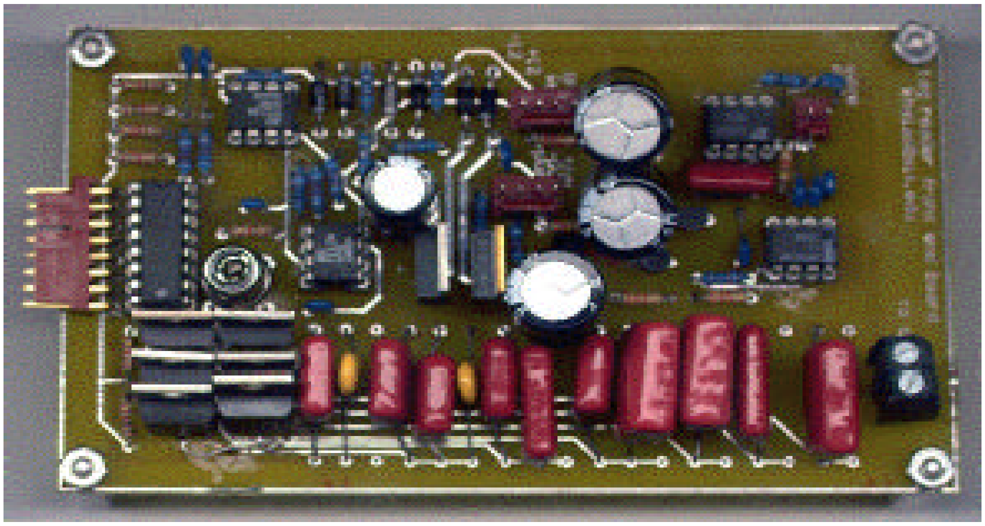


Figure 3. Triac and capacitor ladder board with driver and front-end circuitry.

store more energy at their resonant frequency. This in turn produces a longer, more powerful ringdown, which is easier to detect. LC tags, on the other hand, have a lower Q value, meaning they are less sharply tuned and more lossy, so that they respond a little bit more at frequencies near but not precisely at their resonance, but at resonance they are weaker. The Q value in LC tags is mainly determined by the series resistance of the wire in the circuit, which is not always easy to control.

Another factor is ease of tuning. LC tags have the advantage here because inductors and capacitors for these tags can be selected in a multitude of values on any electronics supply shelf. This means that LC tags can be created to resonate at any arbitrary frequency. Magnetostrictors, in contrast, need to be tuned by cutting the metallic strip to a different length, in order to affect its mechanically-based resonance. This cutting process is fairly imprecise and leads to difficulties in tuning to specific frequencies as well as a potentially significant loss in Q due to mechanical damping in a suboptimal package, which nullifies the initial advantage of using magnetostrictors to begin with.

For the purposes of implementing our system, we opted to use magnetostrictors because of their powerful responses. The ability to tune tags would play a bigger role in a system with more tags, but for a simple test with few tags, we decided that magnetostrictors would produce better responses, which would make the abilities of our system clearer.

The next step, then, was to create a receive circuit for our reader that could detect these tags after excitation.

2.4. Implementation of ringdown reader receive circuitry

Given that the transmit circuitry transmitted pings on a specific set of frequencies, our receive system had to be designed to listen for a signal returned from the tags at their respective resonant frequencies. Furthermore, in order to meet our additional specifications, the system had to be able to determine the magnitude of a returned signal, in order to produce proximity information for the tags.

Once the transmit system finished its ping, any tags in range of the coil would be starting their ringdown responses. The entire receive section of the reader was disabled for the duration of the transmit ping, and continued to stay off for between 100 and 350 microseconds from the end of the ping in order to allow the transmit system to quiet down. After this, the receive circuit was activated and began processing the incoming signal from the coil. Any signal detected by the coil had to be amplified so it could be used; we used a differential amplifier across the coil for this task. As the driver circuit (with the resonator capacitors) was effectively disconnected from the coil via a transmit/receive (T/R) diode switch, the receiver front-end response was broadband. Accordingly, the frequency of the receive signal had to be identified, to determine whether it was identical to the frequency sent out in the ping. Finally, the total power of the response had to be determined in order to infer the proximity of the tag to the coil.

As seen in the schematics in Appendix 1, amplification of the signal was accomplished using a series of filters and operational amplifiers, producing an output that could then be sent to a quadrature demodulator in order to bring the received signal directly to baseband. This had the effect of determining the magnitude of the received signal that was in-phase with the original ping signal, and also the magnitude of the received signal that was ninety degrees out-of-phase with the original ping signal. This was necessary because the ringdown responses of the tags could be subject to dynamic and unpredictable phase shifts due to variations in the tags' resonant peaks from small mechanical differences between tag packages.

The demodulated signal was quite narrowband, i.e. it contained mainly the component of the received signal that corresponded to the expected transmit frequency; out of band noise was removed by simple low-pass filters following the demodulators. The next step, finding the total power of the received signal, could be accomplished by taking the square of each of the quadrature components and then adding them. This was all done in hardware via an analog multiplier the squared the baseband signals, followed by a

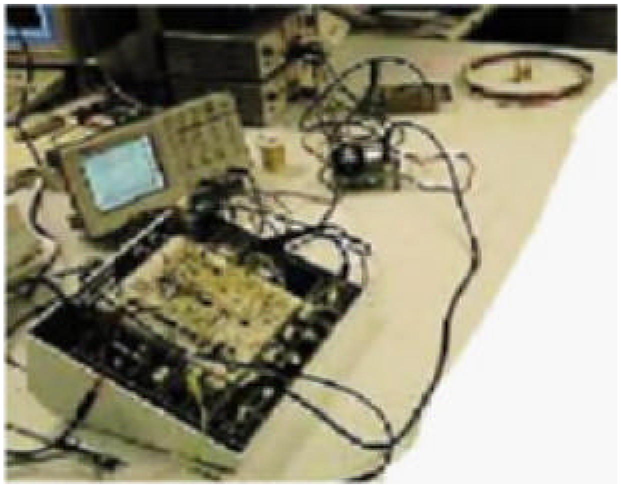


Figure 4. The completed functioning prototype ringdown reader setup.

summer that added them. A gated integrator accumulated the power signal for 10 milliseconds after the transmit ping, producing a single voltage whose magnitude corresponded to the net magnetic coupling, and hence proximity and inclination, of a single tag relative to the coil. Between the 1.5 millisecond ping time and this 10 millisecond accumulation time, along with the wait times in between states, the completed system thus could take around 12 milliseconds in the best case to finish detecting one tag.

With the receive circuitry in place, we were able to finish off the system by taking measurements, setting up a system to digitize the integrator's output voltage, and developing a simple demonstration application to test its abilities. A picture of the completed system on our lab bench is shown in Figure 4. A series of voltage traces showing the performance of the system is shown in Figure 5.

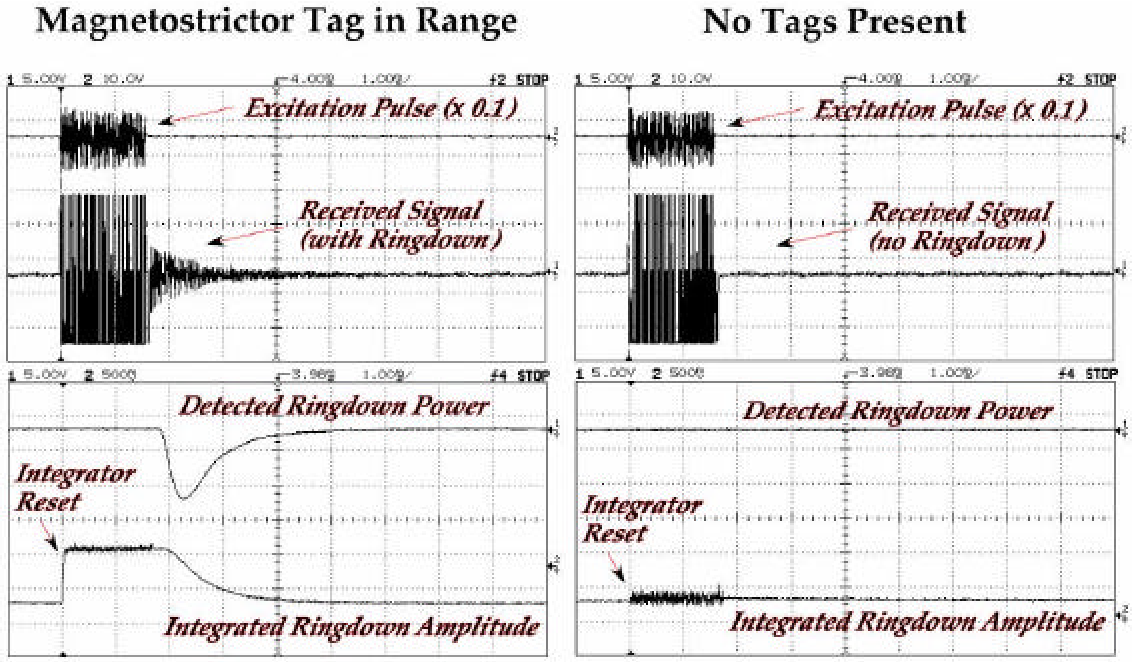


Figure 5. Signals from completed ringdown reader system.

2.5. Ringdown reader demo application and results

In order to assess the capabilities of the system, a standard National Instruments computer data-acquisition board was used to allow a computer to interface with the reader and receive the finished signals coming from each tag detection. A simple Visual Basic application was written, in which three tags were represented by bars on a constantly-updating bar graph display. The height of each bar corresponded to the proximity of the tag to the coil.

The reader was then configured to search at the resonant frequencies of three magnetostrictor tags. By cycling sequentially between the three frequencies, the reader provided constantly updating information about the proximity of each of the three tags. The computer then received this information and updated the bar graphs.

The end result was reasonably encouraging. The bar graphs corresponded accurately to the proximity of the three tags. Each individual signal tended to be affected by noise, but the time-average of each signal was in very good correspondence to proximity, detecting magnetostrictor tags up to 1.5 feet away using a single 1-foot diameter reader coil. However, the speed of the system was not quite fast enough to enable a real-time interface; because our system performed one read in about a tenth of a second, including delays for DAQ card commands, tuning the reader, and software overhead, it took slightly less than a half second to update all three tags.

2.6. Ringdown reader conclusions

In the end, the ringdown tag reader could successfully detect high-Q magnetically-coupled resonators such as magnetostrictor tags at different resonant frequencies and infer the proximity of a tag to the detect coil, provided the tag was oriented correctly to couple to the coil's magnetic field. In addition, the power that can be transmitted through the reader coil can be fairly large and thus the detect range can be relatively good. Because the tuning of the coil can be changed dynamically, the reader is also effective at transmitting efficiently over a range of frequencies, making it possible to detect multiple tags without suffering from static, unchanging coil properties.

The timing in our system, while reasonable, presented several problems for real-time interaction with multiple tags. Because of considerations such as the interval needed for a single tag to be excited and ring down, detecting each tag took about 12 milliseconds. However, in our system we took approximately three reads per tag in order to compensate for the asynchronous nature of the connection between the data acquisition system on our computer and the tag reader, as well as for the time necessary for each section of the circuit to settle into a new frequency setting. Taking a time average measurement over multiple reads also led to improved accuracy, but also required correspondingly longer intervals.

Several disadvantages of this system became quickly evident as a result. First, the system would be difficult to scale to larger numbers of tags. Supposing that we can achieve the feeling of real-time interaction using about twenty detections per tag per second, ten distinct tags would already require about two hundred reads per second. Even using an optimized reader with a decreased detect time, the intrinsic timing considerations inherent to this technique would make it impossible for it to update at this rate.

Furthermore, each tag's frequency has to be preprogrammed into the hardware of this system. The capacitors in the capacitor ladder have to be chosen to enable retuning of the coil assembly to the proper frequency for each frequency range, and the hardware itself must ping at the specific resonant frequency of each tag. This leads to more difficulties because of the possibility of frequency drift and inaccuracies, both in the tags and in the reader. Our device was especially prone to this because we used an open-loop analog voltage-controlled oscillator.

The frequency limitations of magnetostrictor tags were another issue. Although they can be manufactured by today's vendors (e.g. Sensormatic Corp.) at frequencies ranging from 55 to 65 kHz, they are commonly available only at 58 kHz (EAS tags) and 60 kHz (for industrial applications).[14] One can manually cut the magnetostrictor strip inside a tag package or change its magnetic backing to adjust its frequency, but these techniques are imprecise and awkward at best, presenting an additional complication in coordinating the various frequency-dependent sections of the system. The maintenance of such a setup, given ten or more tags, could then be nearly untenable.

Therefore we concluded that although the ringdown reader has an advantageous detect range and good power transmission abilities, we needed to examine other approaches for our real-time, multiple-tag detection purposes before simply trying to further optimize the ringdown reader for our tangible interface applications. Concepts of the sort that we applied to our reader have been used in subsequent work by Fletcher et al.[15] However, for our purposes a system that could quickly read multiple tags without prior knowledge of their precise resonant frequencies would be more suitable for our purposes, and to this end we began examining tag reader designs that continuously “sweep” through a given frequency range, instead of pinging on multiple frequencies discretely.

Chapter 3

Swept-frequency tag readers

Because our prototype ringdown tag reader proved difficult to adapt to larger numbers of tags, we began examining another approach to reading tags. The concept behind swept-frequency tag reading involves passing through an entire range of frequencies in one continuous sweep. This is in contrast to the ringdown reader, which pings at separate, discrete frequencies. As a result, the swept-frequency tag reader uses a different set of operational principles, places other constraints on the system architecture, and as a result has numerous advantages and disadvantages over the ringdown reader.

When a resonant tag receives a signal at its resonant frequency, its circuit draws upon the energy of the reader coil's transient field to produce its sympathetic oscillations. A ringdown reader would listen for the decreasing oscillation after excitation in order to detect the tag. However, a swept-frequency tag reader uses a different principle: When a tag enters the reader's field and is exposed to a magnetically-coupled signal at its resonant frequency, it pulls current from the reader coil. This energy drawn from the field causes a slight, brief, but detectable change in the perceived properties (e.g. inductance) of the coil, which manifests as a dip in the voltage or current being passed through the coil.

A coil transmitting at a constant frequency will only see a change in the coil impedance as a tag with a resonance at that frequency is brought into the field; the further away the tag, the smaller the tag's response will be. Tags resonant at other frequencies will have little to no impact on the coil impedance. By sweeping the frequency instead of leaving it constant, however, any tags in the field produce sharp alterations to the coil properties when the sweep reaches each of their resonances. These changes can be easily emphasized and detected via simple analog signal processing.

As a result, a tag can be clearly detected by sweeping through a range that includes its frequency. This method of tag detection has also been applied to several commercial EAS systems that use LC tags.[16] Most of these systems take advantage of the range of the sweep not to detect multiple tags, since this is currently irrelevant to EAS applications, but rather to robustly detect tags within a small range while tolerating variance in the resonance of manufactured tags. Unlike ringdown readers, the swept-frequency reader that we've designed directly measures the properties of the transmit coil for its detection and so cannot use separate coils for transmit and receive, although other designs can use separate coils to monitor disturbances in the transmit field. Swept-frequency reader ranges also tend to be less than those of ringdown readers, because the tag has to alter the read field itself noticeably, and not just produce its own clear signal without background from the reader. Likewise, since the frequency is continuously sweeping, the dwell time at any particular tag's frequency is limited.

These disadvantages can be easily outweighed by the increase in the frequency tolerance of the system over ringdown readers, and so swept-frequency tag readers have their uses. Specifically for the purposes of detecting multiple tags, an increase in the sweep range to encompass a whole range of tags can bring about the scalability to detecting larger numbers of tags in real-time, which our ringdown reader was unable to produce.

3.1. Design of a continuous real-time swept-frequency tag reader

In order to adapt the swept-frequency tag reading concept to the tangible interface, several constraints must be met. The system must allow for multiple tags; it must be able to determine the continuous proximity of the tag relative to the coil; and it must perform its reading operations quickly enough to enable real-time interaction.

First, the sweep range of the signal being transmitted must span the frequencies of all the relevant tags. In order to detect multiple tags, the receive circuit then needs to be able to measure and distinguish all of the frequencies at which sharp changes in the coil impedance occur, as opposed to just measuring whether or not there is a change at one point in the sweep range. Once this is done, the time during each sweep that a tag is detected provides information about which tag is in the field, essentially converting frequency properties of the tag into timing data.

Next, the system must be able to determine the proximity of the tag relative to the detect coil. The effect of a tag on the coil's measured impedance varies with how strongly the tag couples to the field, as well as how quickly the sweep occurs and how much energy the tag draws from the field. If the sweep rate is fast enough such that the tag appears to be relatively unmoved during any given sweep, we can minimize the effect of the tag's motion on the magnitude of the impedance change. With a fixed sweep rate, the magnitude of this impedance shift is determined by the amount of energy being coupled into the tag to pump its resonance. This in turn is a function of the orientation of the tag and the distance of the tag relative to the coil, both of which determine the coupling of the tag's inductive component to the field of the coil. Thus we can design the system such that there is a relation between the proximity and orientation of a tag and the magnitude of the change in the received signal.

Finally, the reader must be able to run the sweep quickly enough that interactions appear to be real-time. This would involve reading all tags in the system at least ten times per second or higher, depending on the application. Thanks to the nature of the swept-frequency tag reader, each sweep should be able to detect all tags in the frequency range, so increasing the number of tags in the system does not increase the time required to sense those tags. Furthermore, single tags are now detected as soon as the sweep crosses the tag's frequency, since the time taken to listen for the tag's ringdown is no longer necessary.

The first step, then, is to implement a working transmit circuit that generates the proper sweep.

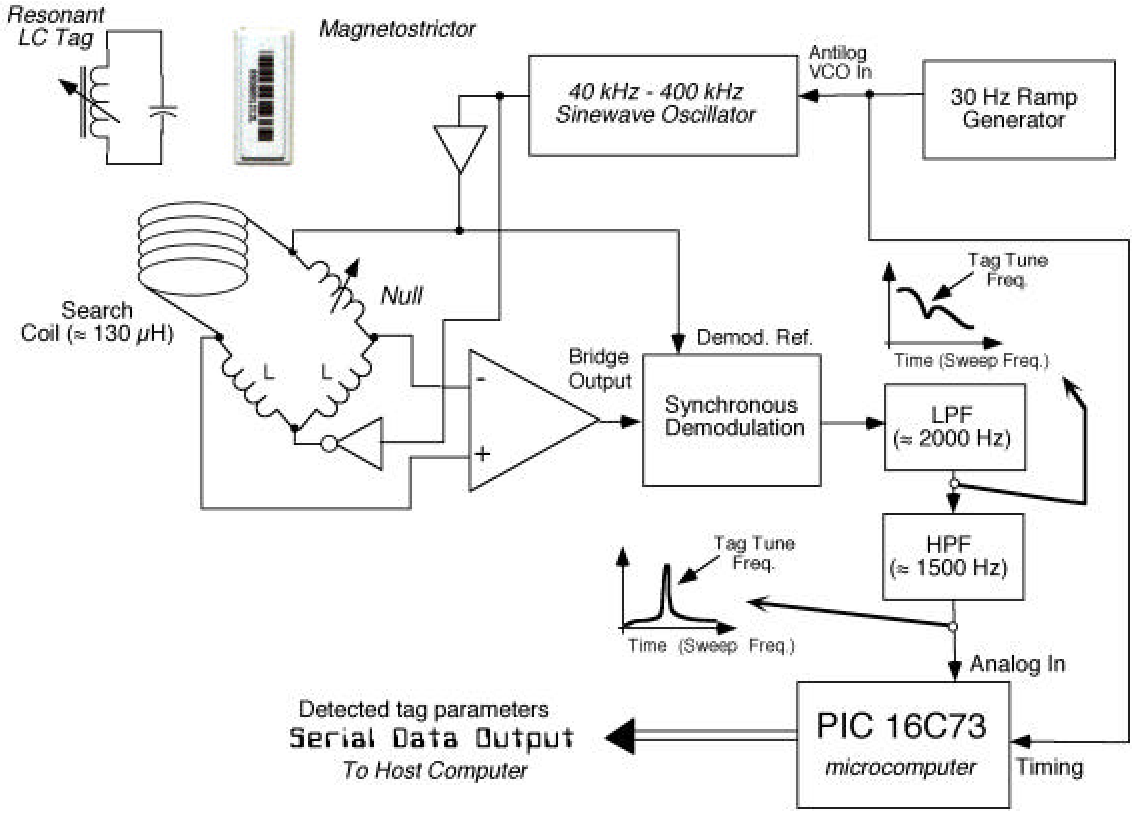


Figure 6. Block diagram of swept-frequency tag reader.

3.2. Implementation of swept-frequency transmit circuitry

In order to sweep through the necessary frequency range, the transmit circuit of the tag reader must be able to produce an appropriate sweep signal. Magnetostrictor tags tend to resonate in the range near 60 kHz, and so we planned our transmit circuitry to sweep from slightly below this frequency to higher frequencies, which are attainable by trimming the magnetostrictor tags to different lengths or by making simple LC tags with appropriate off-the-shelf inductors and capacitors. Our planned frequency range eventually settled to a low of near 50 kHz to a high of near 400 kHz.

In order to produce the sweep signal, we used a voltage-controlled sine wave oscillator. This chip takes as input an analog voltage, which determines the frequency of a sine wave that it outputs. The input to the oscillator came from a circuit that produces a linear ramp signal. This signal simply varied linearly from one user-adjustable voltage to another, and thirty times a second it would be reset to the original voltage. The ramp itself was generated using a 555 timer chip and a single-transistor current source.[17] The essentials of this system can be seen in the upper-right section of Figure 6.

Two factors led to modifications to the input ramp over several revisions of the system. First, it was eventually found that the ringdown of the magnetostrictor tags led to extra artifacts in the detected signals for short periods of time after the frequency of the tag had already been passed. We surmised that this was due to nonlinearities in the magnetostrictor responses. By using LC tags, which have smaller ringdown responses, at the higher frequencies, these false responses could be isolated to the lower end of our tag frequency spectrum where the magnetostrictors resided. A fortuitous result from reversing the sweep to start at a higher frequency and move to a lower frequency also led to these false artifacts having less effect on the responses from other tags by allowing some of the artifacts to take place after the end of the sweep.

Next, the responses of tags at lower frequencies tended to be weaker than the responses of tags at higher frequencies. We hypothesized that this was because a linear sweep spent the same amount of time at a given high frequency that it did at a given lower frequency, but in any given time period there are fewer cycles of a low-frequency signal than there are of a high-frequency one. The solution, then, was to switch to using an exponential sweep, i.e. a sweep that varied less quickly at lower frequencies than at high frequencies. By spending more time sweeping at lower frequencies, more cycles of those low frequencies were produced, giving the tags a more adequate time to respond to the field. The exponential voltage driving the voltage-controlled oscillator was attained by charging a capacitor when the 555 went high and letting it decay as a normal RC circuit.

The end result in our final system was an exponential sweep from high to low frequencies. The resulting sine wave was then passed through a pair of fast operational amplifiers, buffered through push-pull power transistor drivers. These drove the transmit coil differentially, at up to 40 volts peak-to-peak, producing the swept-frequency magnetic field used to detect the tags.

Once the sweep was being transmitted properly, the tags to be used for this system had to be selected.

3.3. Selection of tags for swept-frequency tag reading

As with the ringdown tag reader, the decision to choose one type of tag over the other also came up with the swept-frequency tag reader. Because the two systems use different principles for tag detection, the factors to consider were slightly different.

In a swept-frequency system, the Q value of each tag takes on a different significance; the sharpness of the resonance of the tag now determines how much frequency space in the sweep range the tag occupies, in addition to the strength of the response. Also, ringdown can now become a detriment, because the time spent in one tag's ringdown can interfere slightly with the measurements of other tags that come soon afterwards in the sweep, especially if that tag has nonlinear properties that can produce multiple frequency peaks.

The magnetostrictor tags, with their sharp frequency responses and high Q value, thus took on the advantage of having smaller footprints in the sweep range, which produced the possibility of fitting more tags into any given sweep. However, their strong ringdowns led to interference between tags, and the difficulty involved in tuning them became a much greater issue now, given that the swept-frequency system would be able to read far more tags in real-time than the ringdown reader.

LC tags, on the other hand, with their lower Q values, took up more space in the frequency sweep, but their responses were still thoroughly detectable; in fact, by selecting larger inductors, the tags could couple more effectively to the field of the coil, an effect that helped significantly here, since our sensitivity was somewhat lower than for ringdown tag -reading. The flexibility afforded by being able to select the components of each tag also meant far easier tuning. The relative weakness of the ringdown furthermore decreased interference levels between tags, although, with our analog signal processing, a strongly interacting tag could still suppress the sensitivity slightly to tags at adjacent frequencies. However, fewer LC tags could be fit into any given range due to the width of their responses.

In the end, LC tags provided the most convenient benefits for the purposes of swept-frequency tag reading. However, one or two magnetostrictor tags were also used in the system, left at their original, relatively low frequencies, in order to avoid difficulties associated with tuning them. As discussed in the previous section, adjusting the sweep to hit lower frequencies at the end of the sweep rather than at the beginning allowed the lingering ringdown responses of the magnetostrictor tags to be less of an issue. Having magnetostrictor tags present in the system made it easy to demonstrate that they could be sensed just like LC tags, only with slight complications.

Outside of the few magnetostrictor tags, the remainder of the twenty or so tags eventually used in the system were LC tags, preferred for their ease of tuning as well as the ability of larger inductors to receive more flux from the coil, increasing the response. Note that at these relatively low frequencies, the LC tags were three-dimensional, bulky objects, due to the large inductor coils, as opposed to the larger-area, flat, higher-frequency tags shown in Figure 1.

Having specified these types of tags, we then looked to implementing the detect circuitry for this system.

3.4. Implementation of swept-frequency receive circuitry

On the receive end of the swept-frequency tag reader, several things must be accomplished. First, the slight shift in the coil properties caused by the presence of a tag at a particular frequency must be detected. Next, this must be separated from the oscillations of the original sine wave sent through the coil and processed into a signal that simply indicates the presence or absence of a tag. Finally, this signal must be translated into frequency and proximity information and sent to a computer for further processing.

As mentioned previously, when the resonance of a tag in the field matches the frequency being transmitted through the coil, the tag will draw energy from the field. This leads to a temporary change in the inductance of the coil, which can be translated into voltage changes using an inductive bridge circuit. The bridge circuit is an assembly of four matched inductors (see Figure 6), of which the sense coil is one. The purpose of a bridge circuit is to provide the first step in measuring the difference in a given property of some kind of sensor, relative to a reference value;[18] the voltage across the bridge is proportional to the difference in current flowing down its two legs. Thus, the other inductors in this bridge circuit serve as references, relative to which the small perturbations to the inductance of the read coil can be measured. When the sweep signal is sent through the coil, changes in the inductance of the coil become changes in the current flowing through that leg, which in turn become changes in the voltage across the bridge. We then used a differential amplifier as the first stage to convert the voltage difference across the bridge into an absolute voltage relative to ground. From here, the signal is passed through a synchronous demodulator in order to bring the bridge signal down to DC (removing the sinusoidal carrier) and filter out any noise components at frequencies other than the transmit frequency. After tuning the sweep and nulling the bridge, the result was a smooth signal with small discontinuities that reflected the presence and proximity of a tag. This signal was then passed through a series of various high-pass and low-pass op-amp filters, which enhanced the small bumps and reduced the slow baseline shift (high-pass), and attenuated noise (low-pass). The resulting output came as close as possible to staying near zero in the absence of a tag and rising towards five volts at points in the sweep at which corresponding tags are introduced. The specific sequence of filters can be seen in the schematics given in the Appendix.

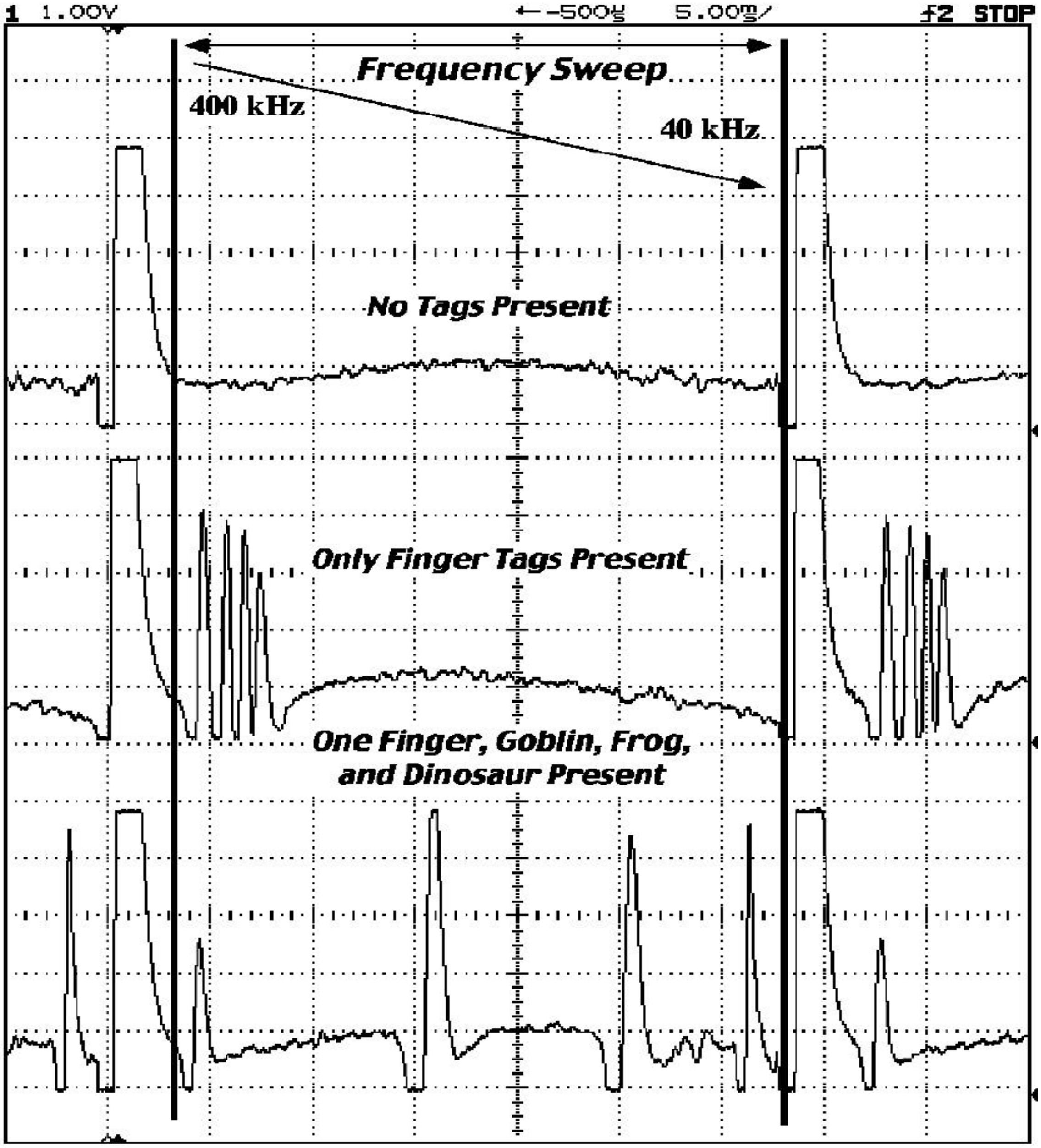


Figure 7. Sample oscilloscope plots of the swept-frequency tag reader signal after the completion of analog processing. Names like “finger tags” and “goblin” refer to objects in which tags have been embedded to demonstrate the system. The large static signals framing the sweep interval signify the boundaries of the sweep in time.

This signal after the completion of filtering can be seen in the oscilloscope plots of Figure 7. This is essentially a frequency analysis showing the response of all resonant objects that approach the reader coil, and could be easily processed using an embedded microprocessor. Thus, we added an onboard PIC microcontroller to measure the voltages

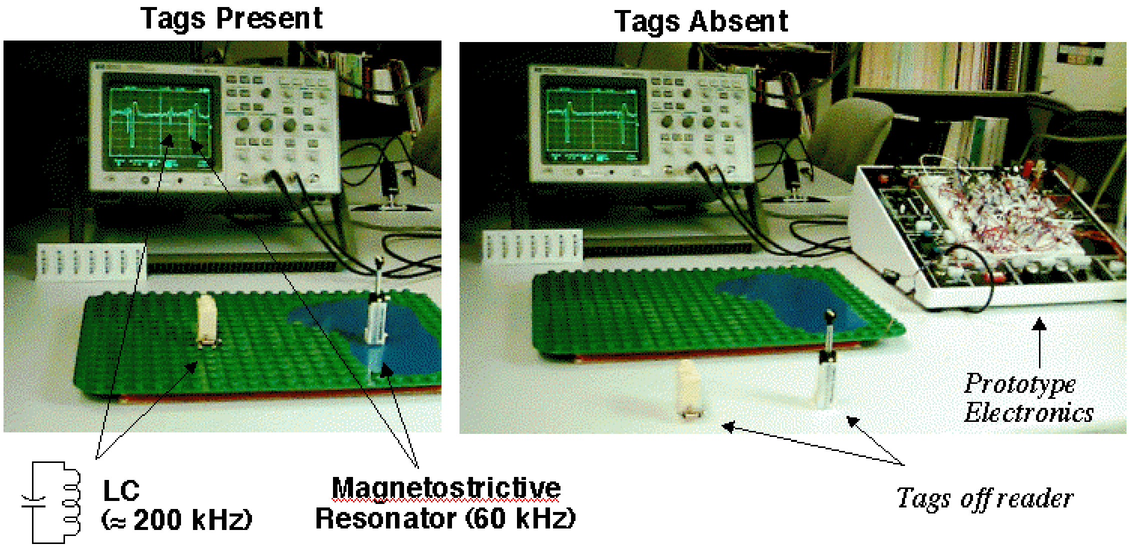


Figure 8. An early prototype swept-frequency tag reader.

of the finished signal and send magnitudes and timing information after each sweep via a serial link to a computer.

An early prototype reader with oscilloscope plots is shown in Figure 8. This system used a coil underneath a Lego board and bricks embedded with tags to demonstrate the basic abilities of a swept-frequency tag reader for applications like interactive games. A later prototype, with improved signal conditioning, is shown with prototype reader boards, coil, oscilloscope plots, and objects embedded with tags in Figure 9; this device was used to develop and demonstrate our initial "Musical Trinkets" application discussed in the next chapter. Figure 10 shows a more finalized version of the circuit board, together with tags and scope traces, that we have used for later versions of the Musical Trinkets and subsequent applications.

3.5. Swept-frequency tag reader results and refinements

As described, we have produced several versions of the swept-frequency tag reader over the past few years, starting from an early prototype reader with very little signal conditioning, up to our current printed circuit board, with selectable sweep types and many

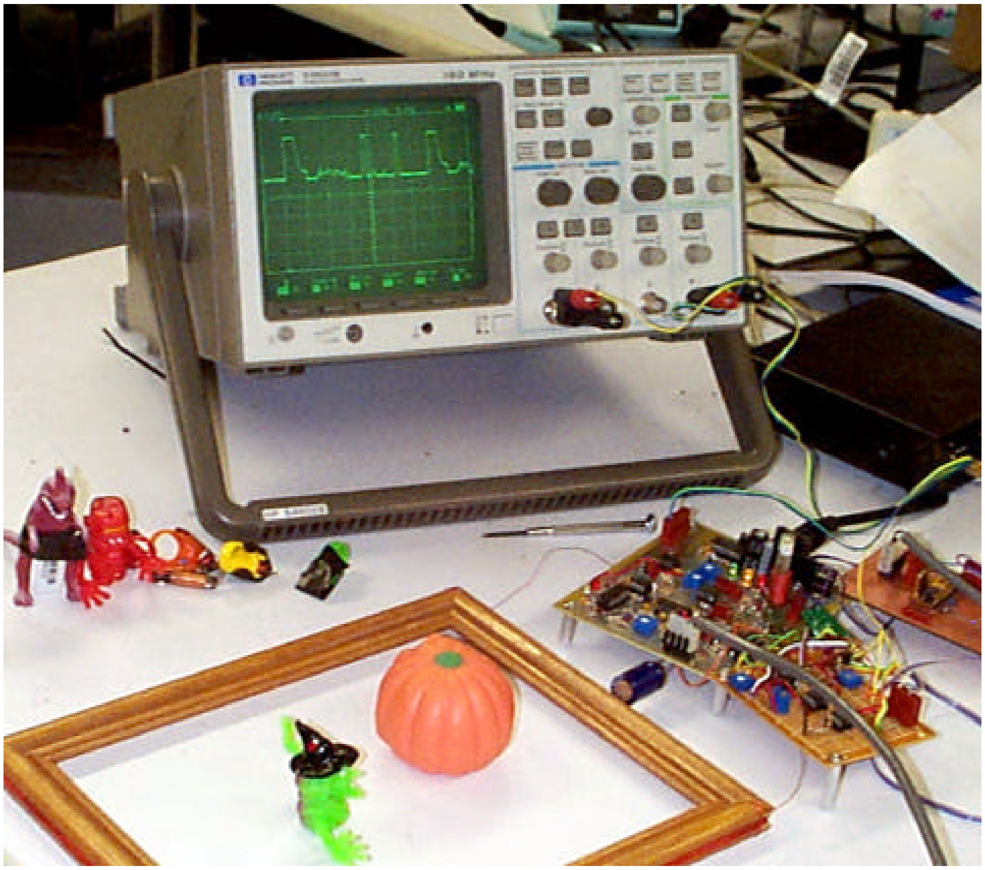


Figure 9. Second prototype of swept-frequency tag reader system.

adjustments. The key advantage common to these readers is that they all can run a sweep 30 times a second, detecting every tag present in the sweep's frequency range and measuring its parameters on each sweep. Because this rate is not affected by the number of tags present in the field, the swept-frequency tag readers demonstrated their superiority for the purposes of real-time multiple tag detection at even their earliest stages.

Depending on the size of the tags, the detection range of our swept-frequency tag reader could extend to a little more than one foot with good reliability. Although not large, this was comparable to the read range of our ringdown reader. Note, however, that the ringdown reader could, in principle, easily be driven with even higher voltages, which combined with its dynamically-tuned antenna would then lead to more improvements in read distance. However, one foot is perfectly acceptable for the desktop-scale interaction with tagged objects that we are exploring.

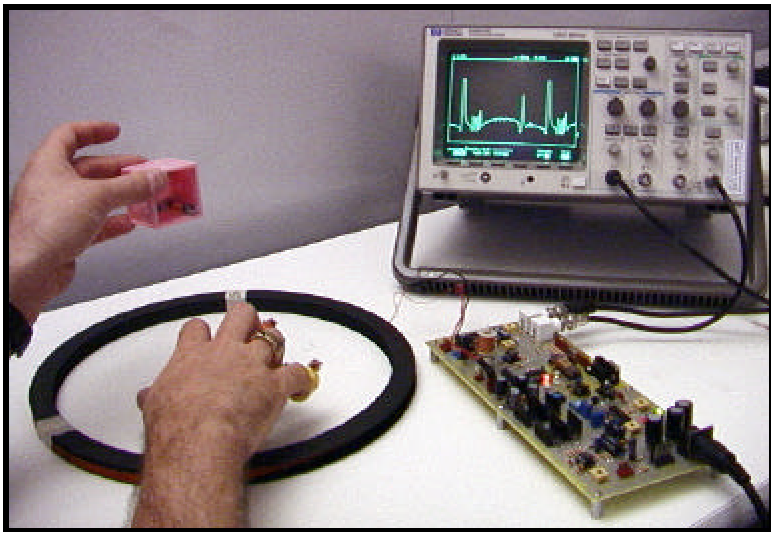


Figure 10. A more mature tag reader. Electronics have been incorporated into a single board, and the ring for the coil was made specifically for this purpose.

The jitter level of our system is quite good across our sensitive region. For example, a typically strong tag was placed in the reader field and held steady for a thousand samples. The PIC measured the integral of its signal and reported a mean of 1685 in 8-bit unscaled units, with a standard deviation of 2.13. This is an average jitter level of around 0.126%, which is easily precise enough for our purposes. Typically, additional background noise is caused by devices that release magnetic radiation in this frequency range, which apparently include CRT monitors and other similar devices. Keeping these noise sources at least six feet away allows the tag reader to retain its usual precision. A commercial implementation of such a system would use higher frequencies, ideally moved away from the dominant sources of environmental noise (again, primarily CRT monitors). These measurements also compare favorably to the ringdown tag readers, which required time-averaging on their proximity measurements because of phase-related difficulties in reading tag responses.

One of the other concerns with the ringdown tag reader was the drifting of frequencies, both within the reader and in the tags themselves. Because of the discrete nature of the ringdown tag reader's detection method, any significant drift in the frequencies being read quickly made the system inoperable. In contrast, the swept-

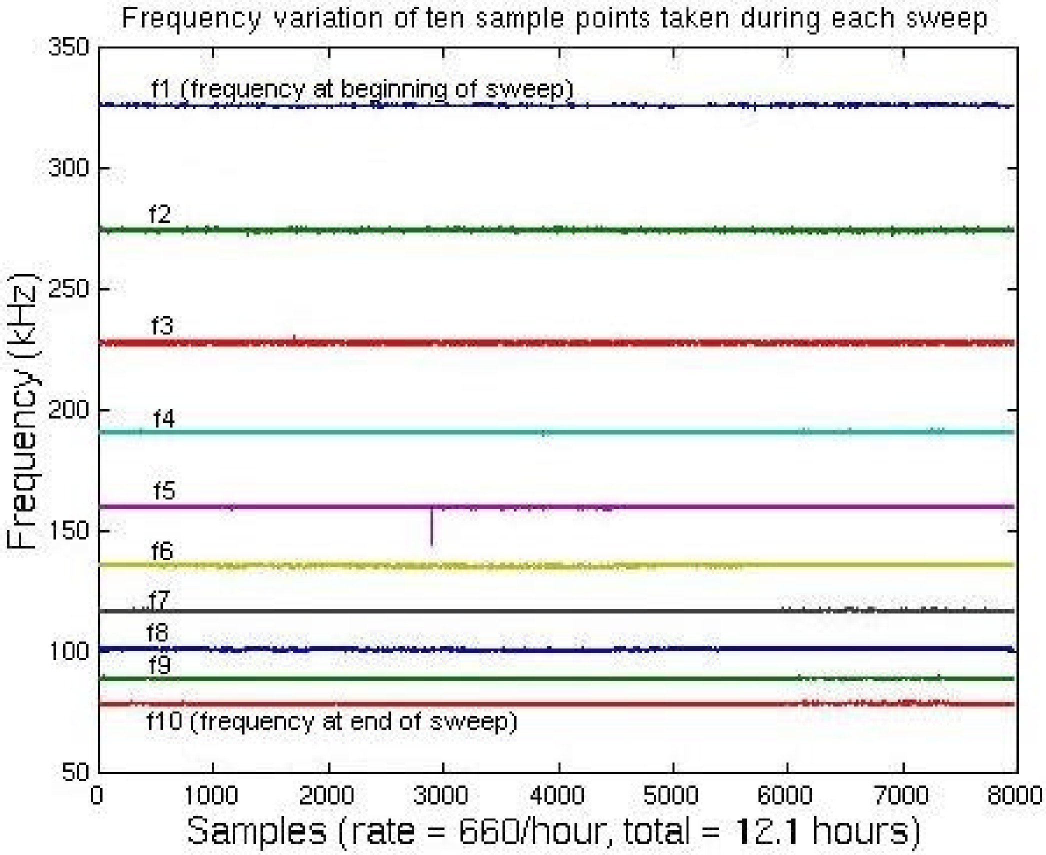


Figure 11. Plot of frequencies at ten sample points during each sweep for a period of 12 hours. Very little fluctuation occurred beyond rounding error. f1-f10 are variables referred to in Appendix B for frequency drift compensation.

frequency tag reader, because it continuously sweeps through a frequency range, ensures that frequency drift has a limited effect on tag detection. However, given enough drift it can become difficult to distinguish one tag from another, as one tag's perceived resonance will overlap the expected resonance of a neighboring tag.

Measurements of voltage being input to the voltage-controlled oscillator, as well as the swept-frequency signal coming out from the oscillator, were taken using the analog-to-digital converter section and the external clock/counter of the PIC, respectively. Timing was monitored using the crystal-locked clock of the microcontroller. The host computer was set up to acquire these parameters periodically and to record them over a period of time in order to determine the frequency drift of the tag reader. Apart from taking these measurements in order to understand the sensitivity of the system, these data can also be used in conjunction with algorithms running in the host computer, while the tag reader is in

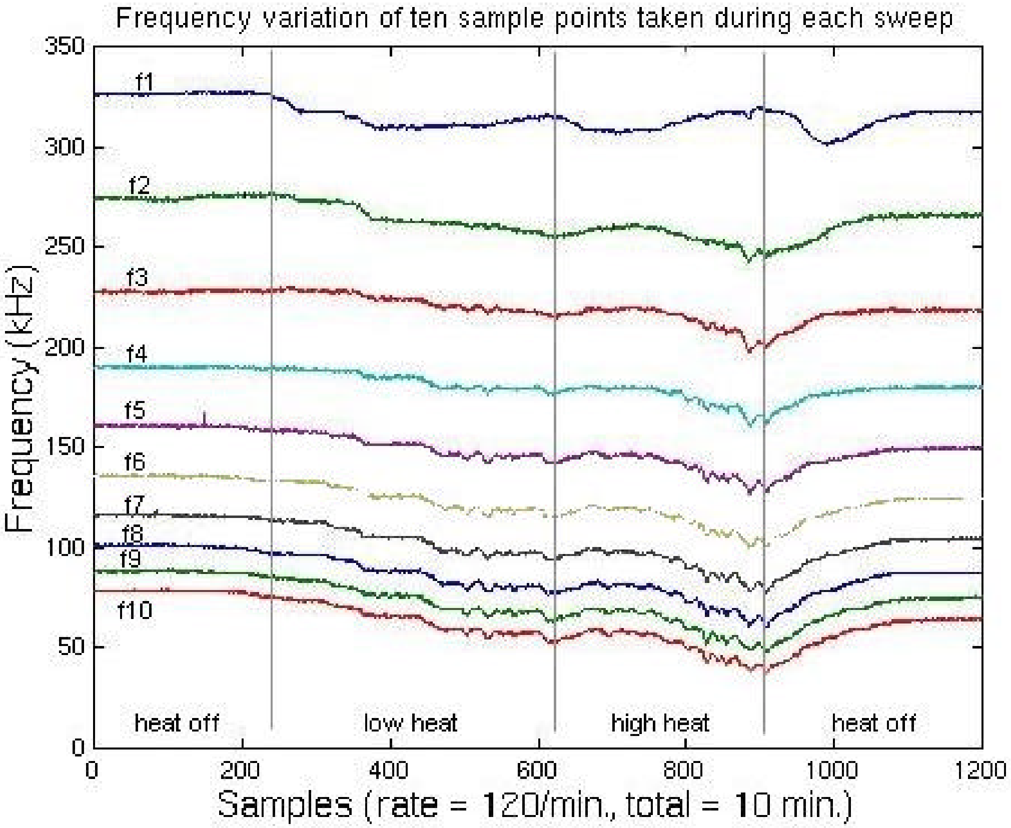


Figure 12. Frequencies sampled at ten points during each sweep while the tag reader was subjected to a heat gun at two different settings, over a period of ten minutes. Notice that on the right side the VCO has still not recovered to its original values on the left; several hours are required for full recovery.

use, to dynamically compensate for such drifting effects and ultimately would allow the system to run with great stability for an essentially unlimited time. The equations necessary to implement such an algorithm are given in Appendix 2.

The measurements themselves provided good insight into the sensitivity of the tag reader. At first glance, we had assumed that time and temperature were the primary contributors to frequency drift in this system. However, monitoring the frequency range of the reader over a twelve-hour period, with the temperature held nearly constant at 75 degrees Fahrenheit by the Media Lab’s central-air system, produced Figure 11, which shows less than a 2-kHz variation from average frequency at any part of the sweep over the course of an entire night. On the other hand, Figure 12 shows the same frequencies plotted over a short period as the ambient temperature around the reader increases at a moderate pace up to 150 degrees Fahrenheit, in order to determine its sensitivity to temperature

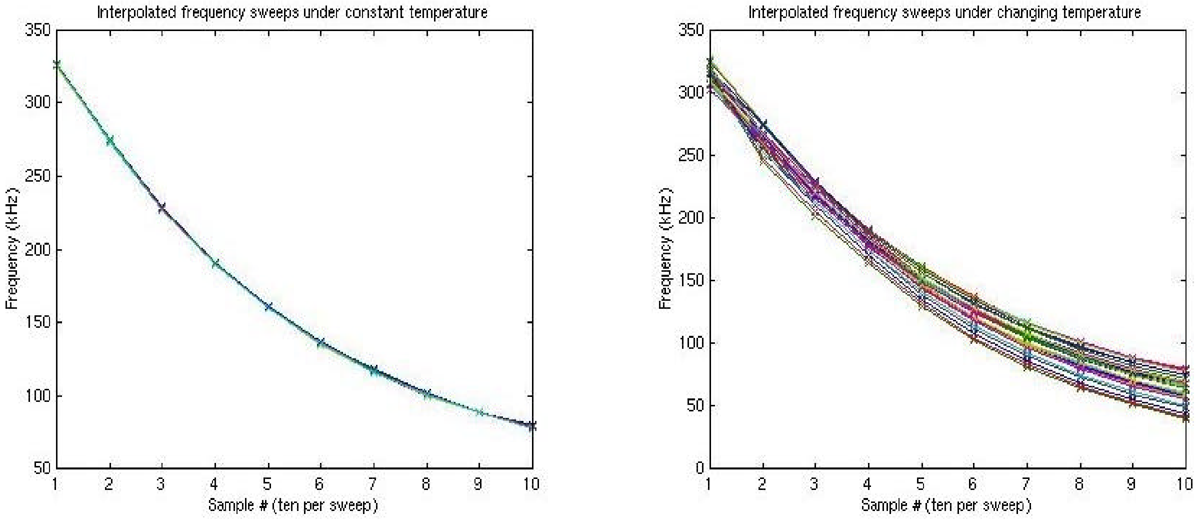


Figure 13. Interpolated frequency curves during single tag-reader sweeps. Left: curves taken during twelve hours of constant temperature, all plotted atop each other. Right: curves taken during ten minutes of temperature change, plotted over each other.

under conditions similar to those used in the "Musical Trinkets" demo (described below). The frequencies dropped dramatically, and the voltage input to the oscillator increased as well. Further measurements determined that the voltage input to the voltage-controlled oscillator returned to its initial value as the temperature returned to normal, but that the frequency output of the oscillator took many hours to settle back towards its initial state. In total, this leaves the system stability fairly sensitive to increases in temperature; Figure 13 shows the frequency curve for each sweep over twelve hours of exposure to steady temperatures, compared to the curve for each sweep during exposure to heat. Finally, some frequency drift was expected with the starting up of the system, but under stable temperature conditions, frequencies decreased by less than 2 kHz between system start and equilibrium, which while possibly significant, is small enough to fall into measurement rounding error.

Even without the algorithms in place to correct for frequency drift, we expect that when placed in an environment with relatively constant temperature, it will be stable enough for most applications. With the algorithms to correct for the drift, the swept-frequency tag reader will have little difficulty maintaining itself in any reasonable demonstration environment for extended periods of time.

3.6. Swept-frequency tag reader conclusions

After numerous revisions and refinements of our swept-frequency tag reader system, we have demonstrated that swept-frequency tag readers have sufficient stability and robustness to drive real-time tangible interface applications. The drift and jitter levels are not by any measure excessive for our intended purposes. Their superior ability to read multiple tags without compromising their timing, the precision with which they can measure tag proximity to the coil, and their stability all help to make this an ideal candidate for further exploration. Because of our decision to use the swept-frequency tag reader in our subsequent work, applications of the reader are described in a separate chapter.

Chapter 4

Applications of the Swept-Frequency Tag Reader

Further work with the swept-frequency tag reader involved the creation of applications to explore and demonstrate its capabilities. Music- and graphic-based systems have generally proven to be a good way to demonstrate new sensing capabilities, as well as to draw the attention of an audience to sensing projects in order to explore their utility at public venues. As a result, our applications took this direction for the purposes of demonstration. Several video files of these projects in action can be found linked off our group's web page (http://www.media.mit.edu/resenv). Also, the reader and applications have been described in several papers published by our group.[19,20,21]

4.1. Early applications

The early prototype reader shown in Figure 7 provided a simple but effective demonstration of this system. The coil of the reader was placed underneath the Lego board shown in the picture, and several Lego bricks were embedded with both LC and magnetostrictor tags. Although at this stage the setup was still a rough prototype and had not been integrated with a computer and thus into a more complex system, the hardware was able to generate tones when tags were detected, with the pitches corresponding to the frequencies of the tags.

This simple output was effective in demonstrating the capability of the system to detect and distinguish between tags, both LC and magnetostrictor. One particular object had three tags placed within it, oriented along the three orthogonal axes, so that its inclination could also be sensed relative to the coil; this was made possible because the signal strength from a given tag always corresponds to both distance and orientation, where increasing distance and rotation of the tag's inductor out of the axis of the field lines both decrease the detected signal. The other tag had a slider that continuously perturbed the resonant frequency of a magnetostrictor by moving a small magnet placed behind it, hence producing a glissando sound as it was pressed. Both concepts, that of using three orthogonally-placed tags within the same object to detect the orientation of the object, and that of adjusting a tag while it was in the field, were appealing for the purposes of demonstration and were thus used in later applications as well.

Another application of a more mature revision of the swept-frequency tag reader was produced in 1999 by Hiroshi Ishii's Tangible Media Group, also at the MIT Media Lab. Their musicalBottles project,[22] in which three bottles on a table each represented a different line of music, was an effective demonstration of the tag reader as a real-time tangible interface backend, and has been widely exhibited. Figure 14 shows a picture of the musicalBottles project in action. As each bottle was uncorked, a corresponding line of music from Edouard Lalo's "Piano Trio in C Minor" was played. The feeling produced by this system was that a user was "releasing" the music from the bottle by opening it. The mechanics behind the system were fairly straightforward; each bottle had a tag inside it, and each cork contained a ferrite that altered the inductance of the tag, thus changing its resonant frequency. The shift in resonance could easily be detected by the swept-frequency tag reader, which reported this information to a computer for processing.

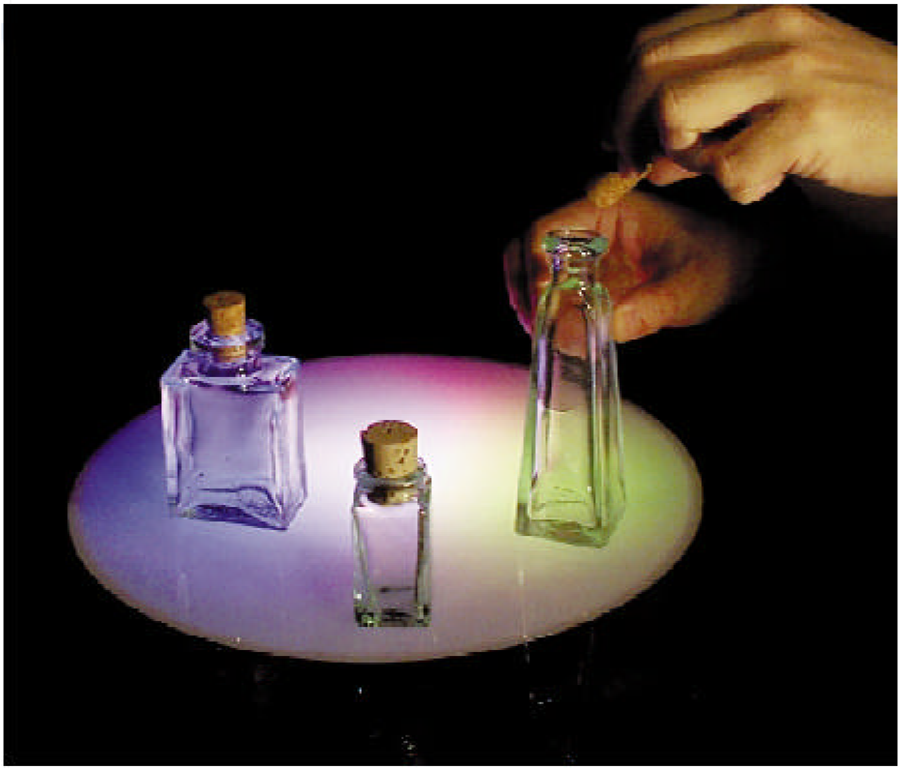


Figure 14. The Tangible Media Group's musicalBottles project. Opening a cork produces a line of music.

The other application for the swept-frequency tag reader produced entirely by our group (Responsive Environments), eventually came to be called "Musical Trinkets." It has a much longer and more involved history, summarized below.

4.2. Musical Trinkets

After numerous revisions, the swept-frequency tag reader evolved into a single printed circuit-board, so it would be easier to reproduce and apply to various projects. In order to demonstrate the abilities of this near-final version, we wrote a more elaborate musical application involving a variety of some sixteen tagged objects.

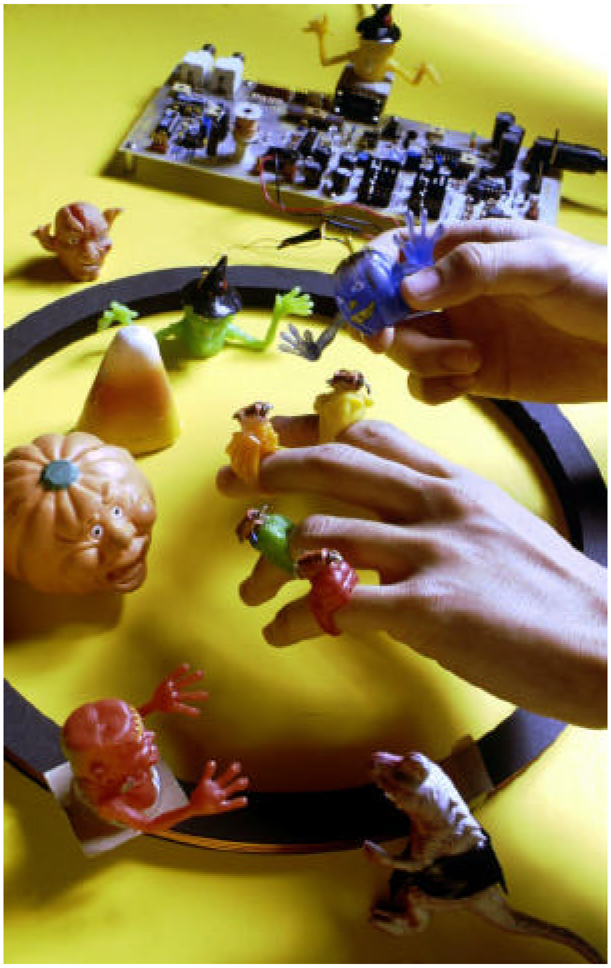


Figure 15. Musical Trinkets, with tag reader, read coil, and objects.

Figure 15 depicts the visual appearance of a fairly mature version of the music-only project. The tag reader was used to identify and sense the proximity (coupled with orientation) of the twenty-or-so tags. This information was gathered by the onboard PIC processor and sent via serial stream to a laptop PC running the music application. The PC sent music commands via MIDI to an Emu ClassicKeys synthesizer, which produced the musical tones that comprised the music. Other MIDI commands were sent to a Lexicon effects processor, which produced various distortions of the output sound.

The basic structure of the music was centered around a melody line and a harmonic chord progression. Five tagged rings worn on the user's fingers allowed the orientation of five fingers relative to the coil to be detected. When the fingers were placed just above the reader coil and rotated to point downwards, melodic notes would be triggered, with a specific major-scale note corresponding to each ring. The effect produced was that of playing a rudimentary keyboard in midair, using one's fingers held somewhere over the tabletop. The speed with which a ring's signal increased, corresponding to the rapidity with which the finger was turned towards the coil, adjusted the MIDI velocity (i.e. volume) of a triggered note. This served to demonstrate the system's ability to determine signal strength, and the ability to play multiple notes at once by inserting multiple rings also demonstrated the reader's ability to read all the tags in real-time without any compromise in timing.

The harmonic chords were controlled by three tagged objects, which happened to be small goblins colored red, blue, and green (which, along with most of the other objects, were all purchased at an after-Halloween sale). As each of these objects was moved closer to the coil, a corresponding bass chord using a sustained instrument voice was faded up. The ability to continuously control the volume in real-time by moving each goblin relative to the coil further demonstrated the notion that the reader could determine the continuous magnitude of the signal in real-time.

Each of the harmonic goblins represented one specific chord in the key of F major. The red goblin represented the C chord (functional I chord), the blue one G (V), and the green one F (IV). Inserting each of these goblins individually produced the corresponding chord. Furthermore, inserting multiple goblins into the field at the same time produced modulations and different chords according to a harmonic state machine, designed to transition smoothly in a harmonic sense between the three primary chords. This state machine is shown in Figure 16. Note that whenever only one goblin was present the state machine always returned to the corresponding state 1, 2, or 3. The chord selected by the goblin objects was used throughout the system; most of the notes produced by the other objects were adjusted to harmonize on the selected chord. This ensured that whatever users did, the resulting sound would not be too discordant.

The other tags in the music system served to either modify, embellish, or add auxiliary layers of sounds to the melody and harmony. For instance, the pumpkin, when

STATE	KEY	IF RED TAG SWITCHES, GO TO STATE:	IF GREEN TAG SWITCH GO TO STATE:	IF BLUE TAG SWITCH, GO TO STATE:
0 (NO TAG)	C	1	2	3
1 (RED)	C	0	5	14
2 (GREEN)	F	9	0	10
3 (BLUE)	G	15	10	0
4	B-FLAT	8	7	6
5	A	2	1	4
7	F	3	4	1
8	E-FLAT	16	3	2
9	G	2	1	4
10	C	11	3	2
11	E-FLAT	10	13	12
12	G	2	1	4
13	F	3	4	1
14	A	3	4	1
15	B	3	4	1
16	A-FLAT	19	18	17
17	B-FLAT	2	1	16
18	B-FLAT	3	16	1
19	D-FLAT	20	20	20
20	F-SHARP	21	21	21
21	B	22	22	22
22	E	23	23	23
23	A	24	24	24
24	D	25	25	25
25	G	26	26	26
26	C	27	27	27
27	F	28	28	28
28	B-FLAT	29	29	29
29	E-FLAT	16	16	16

Figure 16. State machine of harmonies in music system.

brought into the field, produced a high-pitched twinkling rush of random notes whose pitch changed with its proximity. The dinosaur, which had a magnetostrictor attached to it, served as a switch, which triggered an instrument voice change in the harmony and melody lines. The proximity of the foot object caused a continuous pitch bend in the harmony line moving smoothly down by up to an octave while transposing the other lines discretely down by a fifth. The pig object added proportional amounts of vibrato to the melodic and auxiliary lines as it approached the reader.

Several other embedded objects in the system demonstrated slightly more complex interactions. The cube and eyeball each contained three tags, placed at right angles to each other. Because of the correlation between coupling of the tag to field strength and the orientation of the tag relative to the field direction, these right-angle arrangements could yield information about both rotation of the object and distance from the reader, as in the first application with the Lego blocks. The proximity of the cube was used to fade up a low-pitched chord played with a "droning" voice, which then was pitch-bent up and down depending on the orientation of the cube. The eyeball's proximity and orientation was used to adjust the magnitude of various distorting effects on the Lexicon unit.

Perhaps the most interesting user interaction came with the adjustable tag. As discussed in Section 1.2, one of the possibilities with EAS tags is that their resonances can be distorted by various factors, and this shift can be sensed by the tag reader as a change in frequency response. In this system, one of the tags was designed such that the coil of the inductor could be stretched out by the user. This in turn changed its resonant frequency, and the distortion could be monitored by the system. Musically, inserting the tag faded up a choral sound, and stretching the coil produced an additional orchestral string sound on top of it, with the amount of pull on the coil proportionally adding in the second voice. In the final version of this system, the user was able to pull a ferrite core up and down inside a fixed coil, a much more robust interaction than pulling at the coil itself.

The music system was shown at numerous conferences and demonstrations both at MIT and around the world and was well-received by attendees, who were usually impressed by the unique interactions enabled by the system. The music provided intuitive audio feedback, and users had little difficulty learning to use the system and enjoy it. Eventually, the thought came up of proposing the system for exhibit at SIGGRAPH 2000,[23] a very well-attended conference specializing in computer graphics and interaction technologies, held in July of 2000. Because of the heavily graphical flavor of this conference, the next step with this application was to develop a graphical layer of content for the tag reader.

4.3. Graphical extensions to the Musical Trinkets project

For the purpose of adding graphics to the system to enrich the interaction, we decided to go with simple colored shapes on the screen, on the grounds that adding complexity to the graphics would distract from the interaction. Using OpenGL with simple flat single-color polygons in three-dimensional space gave the requisite simplicity while leaving plenty of parameters to adjust with the interaction.

The graphical display was designed to start with a flat gray disc, and other objects would either alter the disc or move around the disc. Triggering the melodic rings produced appropriately-colored falling balls that bounced off the disc. Adding the harmonic goblins faded in red, green, and blue triangles which acted a little like spotlights in that they also added their color to the central disc.

The pumpkin launched a swirling pattern of small diamonds rotating about the central disc, whose radius varied with the tag proximity; the pig caused the melodic balls to shimmer; the dinosaur caused the central disc to flip over; and the foot moved the perspective further away from the central disc, effectively shrinking the view. The eyeball added a floating, spinning eyeball image to the display that moved as the physical eyeball was manipulated, and the cube rotated the entire point of view as it was turned.

For the purposes of demonstration, the stretchable tag was replaced by a Pez dispenser, which had an internal spring that stretched when pulling the head of the container in and out. This made for a very intuitive interaction, especially for people who grew up familiar with Pez. Graphically, moving the Pez dispenser towards the coil added a blue-colored disc behind the primary gray disc, and stretching out the head added still another violet-colored disc.

Finally, just before SIGGRAPH 2000, a new triangular-shaped tag was added, which upon entering the field triggered a transient sound, switched between instrument patches played by the stretching of the Pez dispenser, and also switched between the simple graphical mode described thus far and a new graphical mode written by Marc Downie from the Synthetic Characters group at the Media Lab. This mode, based on Marc's "MusicCreatures" program,[24] was a behavior-based system that listened to the MIDI music commands coming from the computer and encouraged the motion of an abstract animated set of "cloud-like" images on the screen in time with key musical events.

Examples of the first set of graphics are shown in Figure 17, where the visual output of our system was redirected for display on a large screen at SIGGRAPH 2000. A similar projection of the MusicCreatures graphics is shown in Figure 18, with attendees using the Trinkets interface visible in the foreground.

The complete graphical display was set up in a specialized custom-built cabinet that served as a narrow table. A projector at the bottom of the cabinet displayed an image on a frosted glass screen on the top surface; this was used instead of other display devices (e.g. CRT or LCD monitors) in order to minimize interference with the tag reader. The reader coil was embedded just beneath the glass screen so users could move objects near the table surface and receive immediate visual feedback from the graphics projected directly beneath the area of interaction. Figure 19 shows the cabinet at SIGGRAPH with users around it, and Figure 20 shows a closer view of the screen, with a user interacting with the graphics. The system was also subsequently installed at the large SMAU convention in Milan, Italy in October, 2000.[25] Figure 21 shows several attendees interacting with the system installed there.

The whole system made for a very intuitive interaction with aural and visual feedback, which many people described as immersive, stimulating, and often relaxing to use. The real-time capabilities allowed for immediate responses and the continuous values yielded a very detailed level of control. The ability to use all the tags in the system simultaneously also made for a great deal of flexibility in the interaction. On the whole, the system was very well-received by the general public, many of whom were intrigued by the novel mode of interaction. People were both fascinated by the Musical Trinkets demonstration itself as well as the potential applications that could be enabled by such a system.

Once the demonstration application was set up and running, and the capabilities of the basic swept-frequency tag reader were clear, it became possible to consider extensions of the system beyond its simple single-axis roots, to explore volumetric tracking applications.

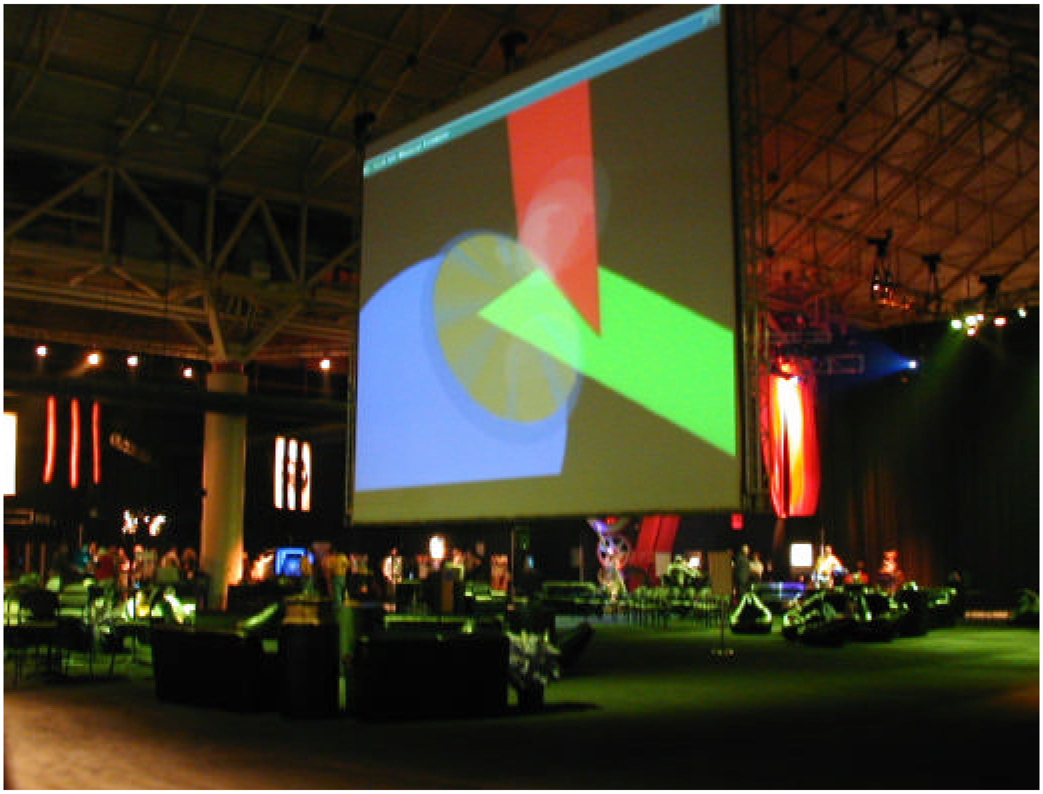


Figure 17. The first set of Musical Trinkets graphics, projected onto a large central screen at SIGGRAPH 2000.

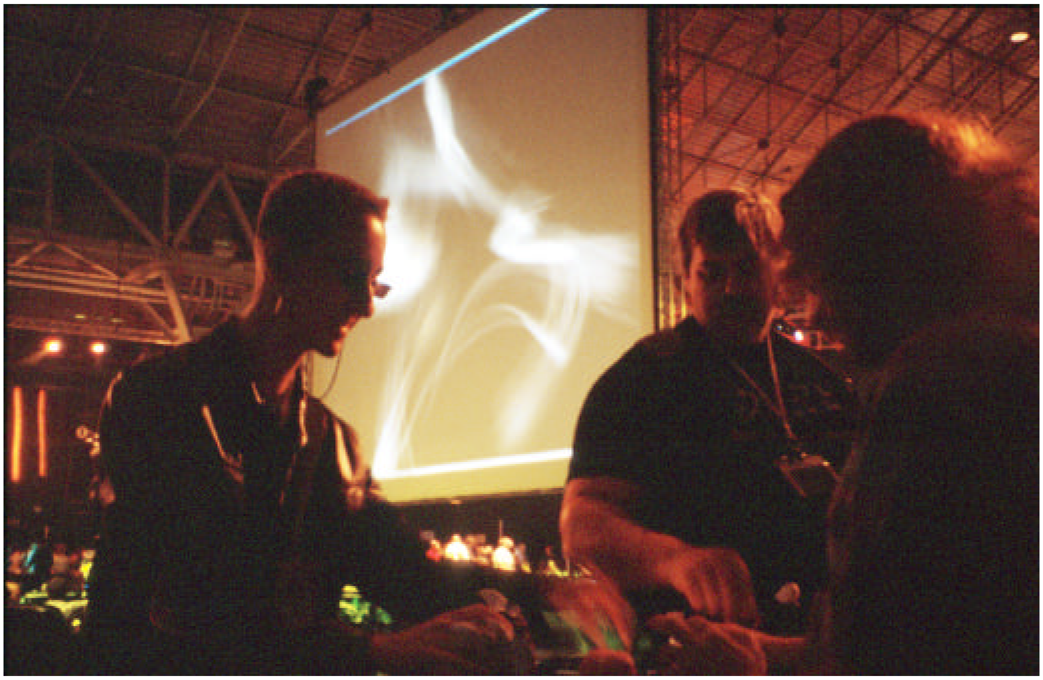


Figure 18. The graphics based on MusicCreatures, projected onto the large screen with users using the Musical Trinkets in foreground.

Figure 19. A small audience gathers around the Musical Trinkets exhibit at SIGGRAPH 2000.

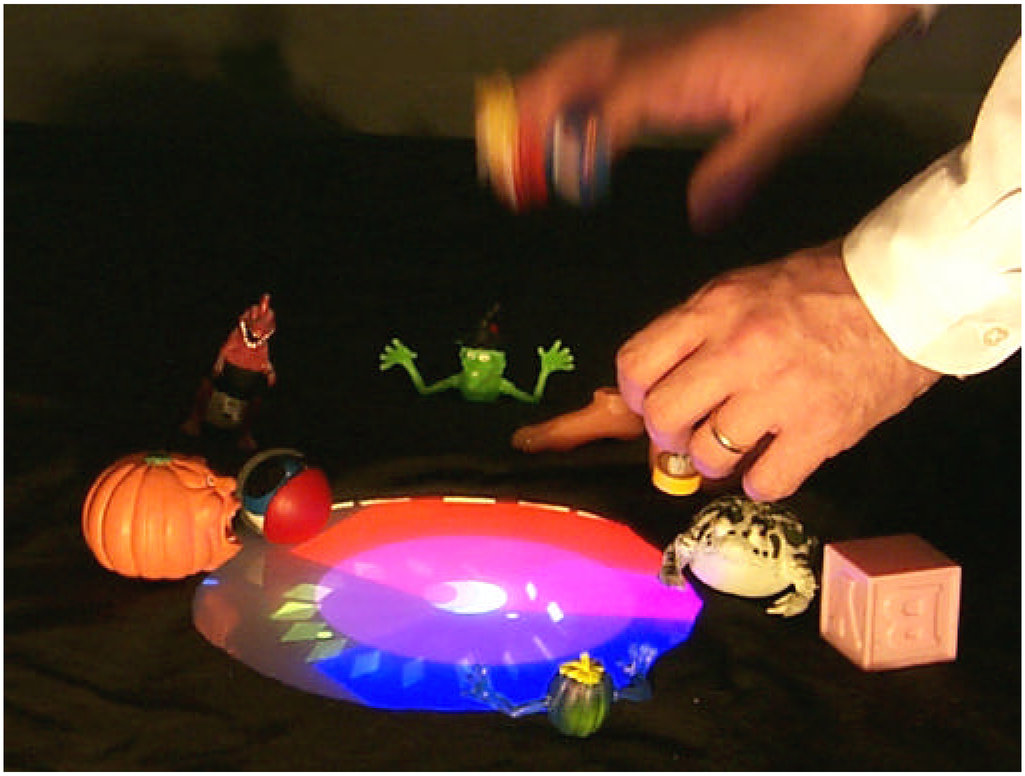


Figure 20. A user interacts with the graphics screen.

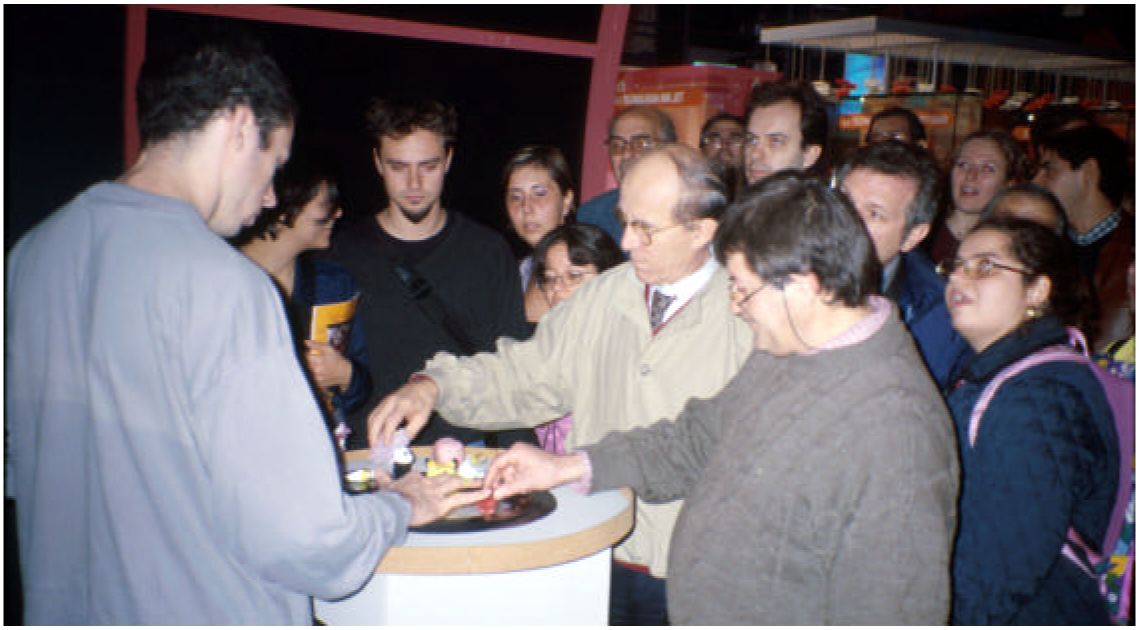


Figure 21. Throngs of onlookers watch the Musical Trinkets project in action at SMAU in Milan, Italy.

Chapter 5

Swept-frequency tag readers with multiple coils

With a single read coil, the swept-frequency tag reader is capable of determining signal strength from a tag in real-time. As mentioned, this signal strength is a function of both proximity of the tag to the coil and the orientation of the tag's inductive component relative to the field lines emitted by the coil.

As discussed in the previous chapter, this in itself can provide a rich experience, but we wondered if the system could be modified or extended in order to provide more precision, more uniformity of interaction, or more information about each tag. Any of these would enable a more interesting interaction for applications that might require anything more than simple gross tag position in one direction relative to a tabletop. By considering the limitations of the current system, modifications were considered and some were implemented to extend the reader's capabilities.

5.1. Limitations of a single-coil swept-frequency tag reader

A swept-frequency tag reader using a single read coil and simple, imprecise tracking enabled a fairly rich interactive experience as demonstrated by the Musical Trinkets application. In the interest of considering extensions to the system, we noted several limitations, which mostly involved uniformity of tag reading and cross-axis coupling with respect to position and orientation of the tag.

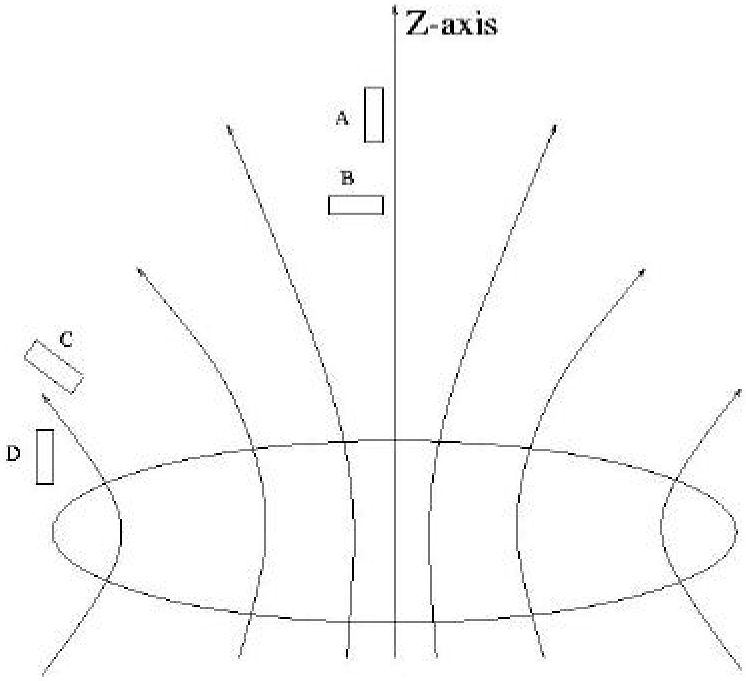


Figure 22. Simple illustration of single-coil field and tag interaction. Tag A is turned parallel to the field lines and so is detected strongly. Tag B is turned perpendicular to the field lines and so is detected weakly. Tag D is turned in the same direction as Tag A, but the field lines there run in a different direction. As a result, Tag D is detected less strongly than Tag C, which is turned more parallel to the field lines there. This is an example of non-uniform interaction, because of the differences in the detection of Tags A and D.

In a uniform interaction, changes in a tag's orientation would have similar effects on the magnitude of its signal, regardless of the location of the tag. One of the issues that prevents uniformity of interaction with a single-coil tag reader is the nonparallel shape of the field lines surrounding the coil. Right at the plane of the coil and on the central axis, field lines run uniformly perpendicular to the plane, but anywhere off the plane and off-axis, field lines begin to diverge, bending in elliptical shapes around the coil as the magnetic flux follows its return path. Because of this shape, tags placed in the field directly over the coil's center are sensed with a magnitude corresponding to their orientation relative to the axis of the coil, but tags not directly over the center have a stronger magnitude when rotated slightly off-axis. Tags directly over the wires of the coil itself are detected most strongly when turned completely perpendicular to the coil's axis. See Figure 22 for a simple illustration of this principle. If this problem could be solved, the response of the system to the orientation of a tag would be more uniform, which would increase the intuitiveness of the interaction.

Thus, the nonparallel field lines cause problems for the uniformity of the interaction, introducing an ambiguity between angle and range depending on the tag's position. Another problem that arises is that the signal strength from a tag increases as a tag approaches the wires of the coil, i.e. towards the outside of the ring. This is because the field is really produced by the induced magnetic effect at the wires wound around the ring, hence the magnetic flux is denser there and decreases with distance from those wires. As a result, it is difficult to determine whether a decreasing signal strength is an indication of a tag moving away from the tabletop or simply closer to the center of the coil area. This is another issue that affects the uniformity of the interaction; in this case, reducing this effect would allow for more precise determination of whether a tag was close to the plane of the coil or not.

Even despite these uniformity difficulties, one more basic limitation was that with a single read coil, only one piece of data about each tag could be generated. This single piece of data, which corresponded to a function of both position and orientation, was in itself already quite useful in producing a novel generic interaction, but moving a tag horizontally around the tabletop, as many of our SIGGRAPH and SMAU attendees wanted to do, produced no interesting effects as the motion could not be effectively detected. Nor could a tag being rotated be differentiated from a tag being moved closer and further from the reader. In the single-coil applications, this was circumvented by using three differently-tuned orthogonal tags in a single object, which gave some idea of the orientation of the object, but still produced no data about the horizontal position.

Because these limitations seemed to center around difficulties associated with having only one reader coil, we turned our attention to possibilities for arrangements of multiple coils.

5.2. Multiple-coil arrangements and Helmholtz coils

The extensions of the swept-frequency tag reader that we expected to produce the most promising results, with respect to the limitations we noted, appeared to come from adding more reader coils to the system. Given an appropriate arrangement, it seemed that multiple coils could improve the uniformity of the system as well as provide more information about the location and orientation of a tag. We realized early on that having multiple coils did not necessarily mean running all the coils simultaneously, and in fact cycling sequentially through a series of coils might lead to a slowing of the read time, but may be necessary in order to prevent multiple coils from interfering with each other. Despite this, for many applications, some degree of slowdown would be a reasonable exchange to enable more precise varieties of interaction.

Several types of multi-coil arrangements were considered. First was the possibility of having several coils placed in the same plane, either side by side or in a grid pattern. By comparing the signal strengths for a given tag, it seemed like perhaps it would be possible to tell which coil a tag was closest to, and from this the horizontal position of the tag could be inferred. Likewise, by driving coils differentially, other field geometries could be created that would provide additional information to resolve rotation, break ambiguities, and determine a more precise location. At first glance, the math might work out similarly to that in Josh Smith's "School of Fish" project,[26] which used an array of electric-field sensing electrodes to accomplish a similar localization task for capacitively-coupled objects. However, because of the nonuniformity of the magnetic fields coming from the coils, and the vector coupling between fields and tags, the math could become unwieldy, and no easily discernable reader coil configuration was derived for determining the precise location of a tag. Thus, this solution promised to yield more information about the whereabouts and state of a tag but without considerable calculation, would do nothing to increase the uniformity of interaction.

Another arrangement that we considered was having a set of three orthogonal coils, meeting at one corner and sequentially transmitting a field along its axis. This would

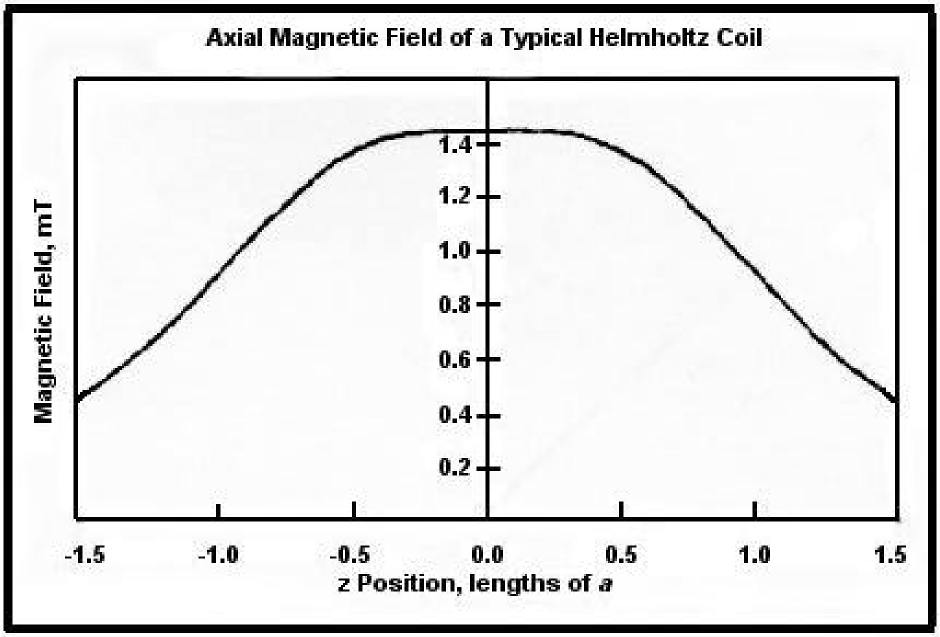


Figure 23. Axial magnetic field of a typical Helmholtz coil, with coils spaced distance *a* apart, where *a* is the radius of a coil.

provide capabilities similar to the single-coil system, except on three axes, giving some degree of spatial knowledge to the measurement. Such an arrangement would give some idea of three-dimensional position, but once again the field shapes were unchanged and the uniformity of the interaction relative to any single coil would be compromised, as in Figure 22.

The dissatisfactions with both of these first two arrangements were the lack of simple field shaping and uniformity. Our next concept, then, was to have two coils facing each other, driven by the same signal simultaneously, in the hopes that this would take the elliptical magnetic field lines of the single-coil system and pull them through two coils, creating a space in-between where the field lines would be straighter and more uniform. This arrangement is commonly known as the Helmholtz coil configuration.[27]

A little research yielded more information on Helmholtz coils. A single-coil system has a magnetic field given by $B_z(z) = \frac{Nm_0 Ia^2}{2(z^2 + a^2)^{3/2}}$, where N is the number of wire turns in the coil, I is the current through the coil, a is the radius of the coil, m_0 is the constant of magnetic permeability and z is the distance from the coil being measured. This magnetic field decreases rapidly with increasing distance from the coil. Helmholtz coils, on the other hand, have fields along their central axis given by:

$$B_z(z) = \frac{Nm_0 I a^2}{2}\left\{\frac{1}{(z^2+a^2)^{3/2}} + \frac{1}{[(2b-z)^2+a^2]^{3/2}}\right\}$$, with 2b being the distance between the coils and other variables the same as above. While the equation looks messy, the field it represents is much more uniform between the coils than in the single-coil system. A plot of the Helmholtz field is shown in Figure 23. Notice that the field strength is approximately constant between the coils when the distance separating the coils becomes approximately equal to the radius of one of the coils.

Because of the improved uniformity of the field in the Helmholtz configuration, we decided to try such a two-coil setup to observe its properties.

5.3. Two-coil Helmholtz arrangement

Using the boards already designed for the single-coil swept-frequency tag reader, we were able to create a simple two-coil Helmholtz arrangement by wiring two reader boards together such that the sweep generation circuitry from one board was used as the input to the drive/transmit circuitry of both boards, with loading by the tag measured separately for each coil as detailed in Chapter 3. A different coil was connected to each board and they were placed parallel to each other approximately one coil diameter apart. A simple block diagram of this setup is shown in Figure 24.

One of our uncertainties with using two simultaneously-transmitting coils was the way in which the tag would couple to each of the coils. Would both coils receive equal signals, would the signals between coils cancel, or would the signal received vary proportionally to the distance between the tag and each coil? Our test showed that the tag coupled with the two coils such that the signal received varied approximately linearly with the proximity of the tag to one coil or the other.

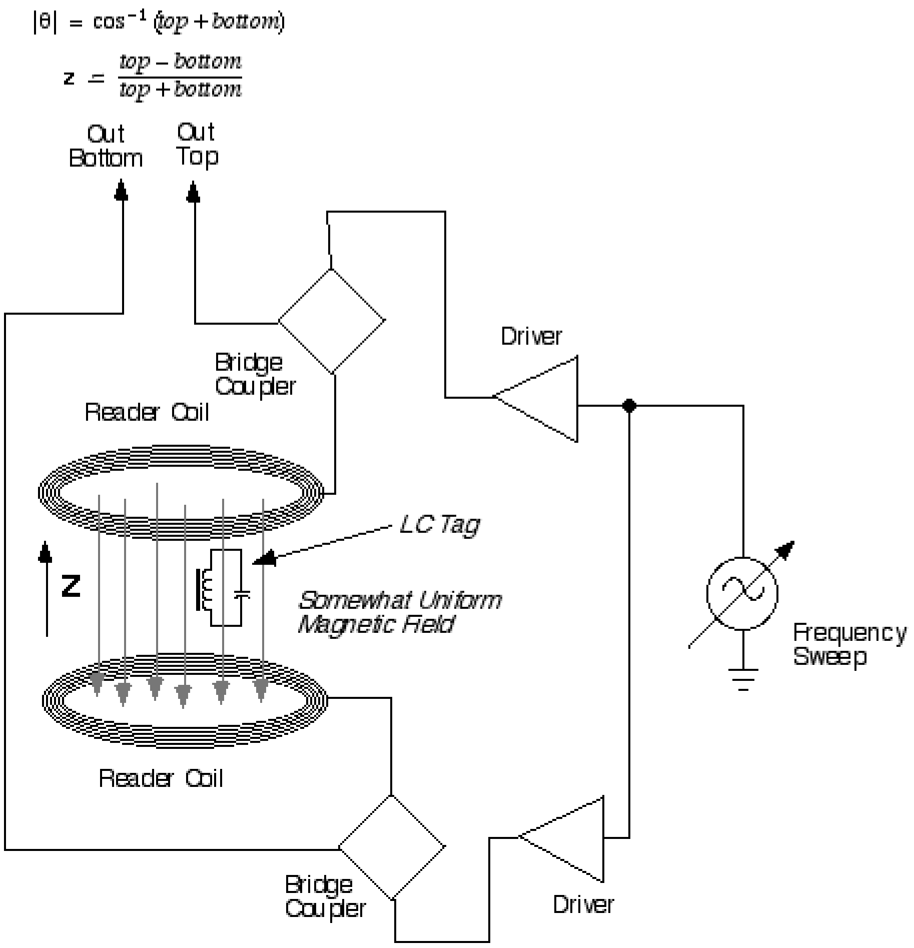

$|\theta| = \cos^{-1}(top + bottom)$
$z = \frac{top - bottom}{top + bottom}$
Out Bottom
Out Top
Driver
Bridge Coupler
Reader Coil
LC Tag
Z
Somewhat Uniform Magnetic Field
Frequency Sweep
Reader Coil
Bridge Coupler
Driver

Figure 24. Two-coil Helmholtz arrangement block diagram.

Field shaping improved substantially as well. This two-coil configuration apparently had the ability to shape the magnetic fields to be straighter and more evenly spaced in a larger region between the two coils, relative to the one-coil setup. The result was that the tag's orientation decreased the signal strength in both coils proportionally when rotated out-of-axis, provided that the tag was not too close to the edges of the coils.

Note that the optimal Helmholtz configuration involves spacing the coils one coil radius apart from each other, but our systems used coils spaced one diameter apart, which tended to lose some uniformity towards the fringes of the field and otherwise provided the same benefits.

From these observations, the results were promising. Inside a sizable region within the coils, the field shaping of the Helmholtz configuration essentially decoupled the orientation of the tag from the position of the tag along the axis between the two coils. As summarized in Figure 24, the position of a tag could ideally be given by the difference in signal strengths normalized by their sum, and the rotation of a tag out-of-axis was a function of the combined signal strength in both coils.

Furthermore, the two coils were transmitting simultaneously, leading to no slowdown in the response time, and also we were now receiving two pieces of information about each tag, determining both position and orientation. The only desired feature missing from this system was the ability to read the position of a tag along more than one axis. Our next system used similar principles to do exactly that.

5.4. Six-coil Helmholtz arrangement

As seen in the two-coil Helmholtz arrangement, it was now possible to use two coils to gather two pieces of information about a tag: position and orientation. Our next hope was to extend this concept to gather six pieces of information about a tag, i.e. position and orientation along three orthogonal axes.

The logical way to create such a system was to have three pairs of coils placed along the outside of a space, each in different directions. The immediately apparent disadvantage to such a system was that having coils totally surrounding a space would limit a user's ability to move completely flexibly in and out of the space. We decided that using the six faces of a cubical frame as coils would be substantially better for user mobility than having six attached circular rings. The edges of the cube were only needed to support the wire of the coils, hence could be made very thin and unobtrusive.

A cube structure was built for this purpose, and wires were wound appropriately around the six faces and each connected to a separate tag reader board. Our initial cube was nine inches on a side, and we soon constructed a new cube out of Tinkertoys that was

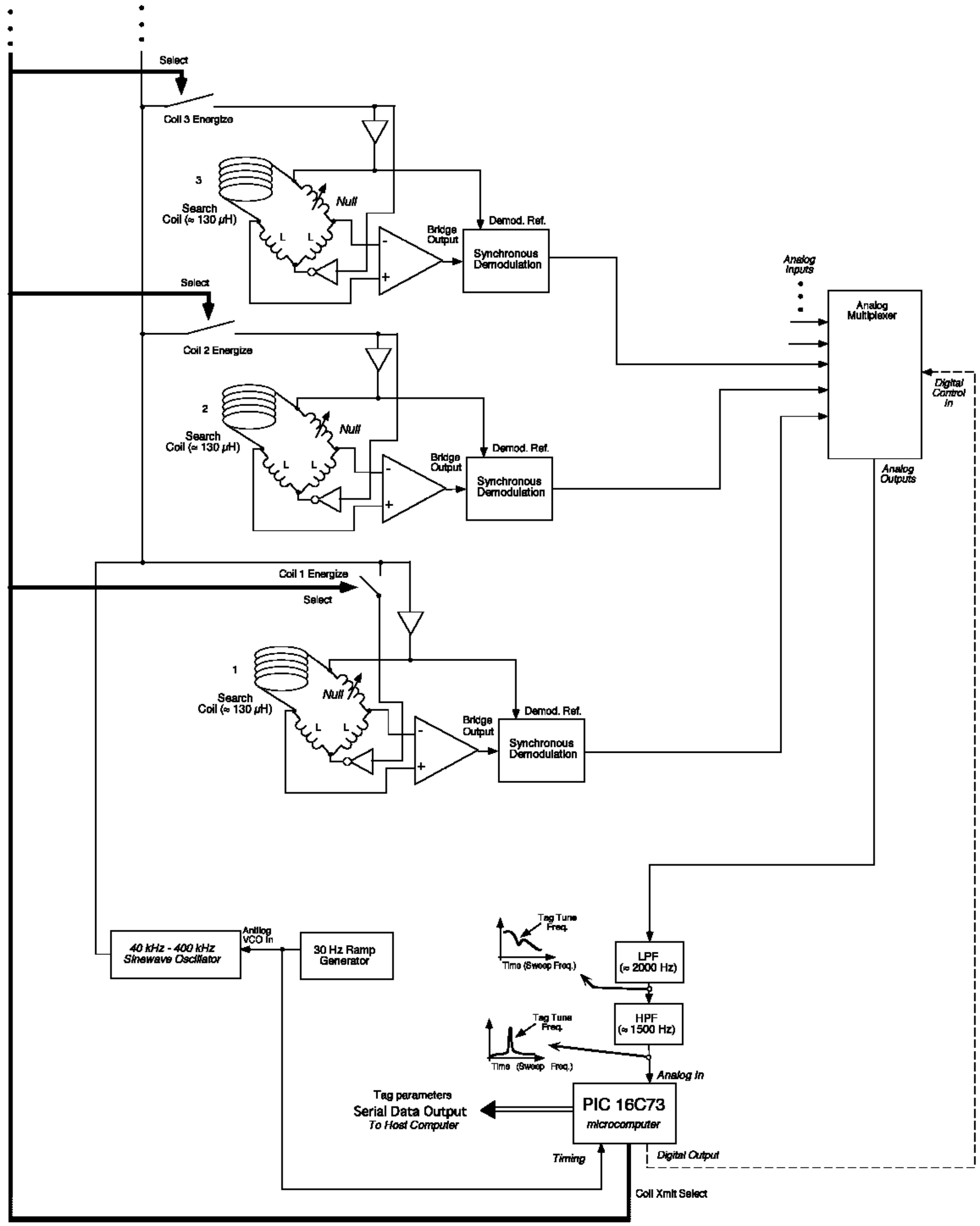


Figure 25. Block diagram of six-coil tag reader system.

twelve inches on each side, with ten windings of wire in each coil, making for an inductance of between 105 and 115 µH, thus matching the inductors in the bridge circuit on the reader board. The next step was to connect the reader boards such that the six coils

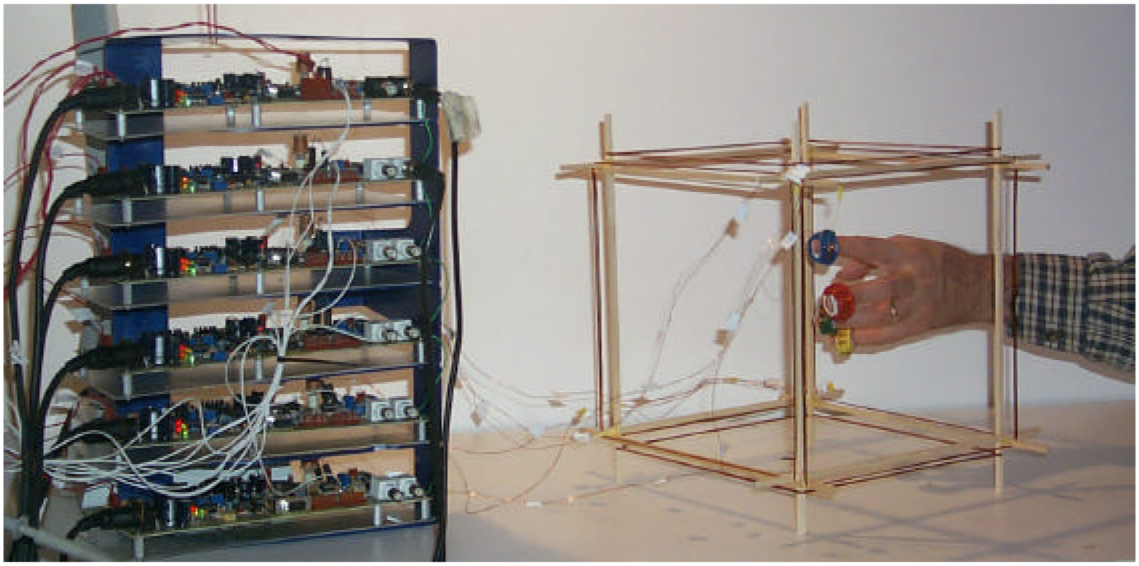


Figure 26. The first completed six-coil cube frame with six tag reader boards.

transmitted sequentially, one opposing pair at a time. This pair-wise reading was necessary in order to create Helmholtz fields between a pair of coils on one axis, while the other four coils were turned off in order to prevent them from affecting the shape of the magnetic field or the resulting signals. By cycling sequentially between three sets of coils, we slowed our reading time by a factor of three over the single-coil case, but we anticipate that many applications of the system would not suffer greatly from this effect. Otherwise, the sweep could be correspondingly sped up with minor changes in the electronics, keeping full updates of at least 30 Hz.

As depicted in Figure 25, the connections between the six reader boards were closely related to the connections of the two-coil system. One board was designated the master, and the sweep generator from this master board was fed through a multiplexer into each of the drive circuits for all six boards. The multiplexer itself was controlled by the PIC chip on the master board, so that every time a sweep was completed the PIC switched the multiplexer to change connections to the next pair of boards. The drivers for all six boards functioned normally and transmitted whenever they received a signal from the multiplexer.

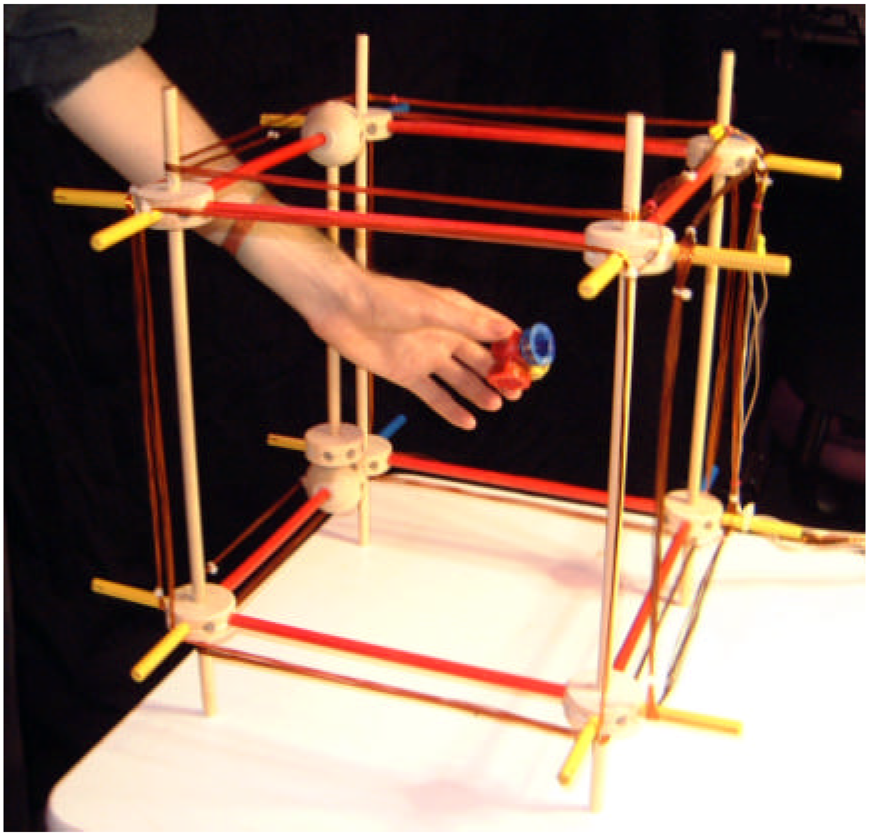


Figure 27. Our final cube frame, along with three-axis tag and attachments for taking measurements.

On the receive end, the signals coming back from all six coils went into their respective boards, and after some analog processing were transferred back through a multiplexer to the master board and its partner. The final filtering stages were performed on these two boards, and the two PIC microcontrollers on these boards were used to assemble the final information and transmit it via two serial cables to the host computer. A picture of the completed system with all six coils around the cube is shown in Figure 26. The newest six-coil cube, in the configuration used to take measurements for the next section is shown in Figure 27.

Having completed the six-coil system, we observed its behavior and created a simple demonstration application.

5.5. Results and application

Initial tests of the six-coil system yielded reasonable results. The behavior observed earlier with the two-coil system was repeatable with all three pairs along the different axes in this system. Differences in the response strengths of individual tags created variations in the reconstruction necessary to determine position and orientation and the precision achievable with each tag; some tags were weak and their signals faded out towards the center of the cube, while other tags were strong and could saturate at the cube's edges. From this we inferred that the system must be carefully tuned in order to achieve the best results. Increasing the analog conversion precision to ten bits of information also improved the precision; this was added in the final version. A dynamically-sampled baseline and smarter thresholding algorithms would further improve the ability of the system to distinguish noise from signal, and has not yet been added; we currently assume a static DC baseline across frequencies, but this is rarely the case (see Figure 7 for example signals and baselines).

Noise levels were also higher for some of the reader boards than for others, which we attributed to variations in calibration settings and soldering connections. Also, because the bridge imbalance did not null as well across the entire decade of sweep used in our other versions of the system, we swept across a narrower range of frequencies, over which the baseline (and thus the null) was fairly flat. This improved the noise response as well. Another potential contributor to noise level is the driver design; our coil driver circuitry essentially holds non-driven coils effectively shorted, which may allow eddy currents to build up and distort the magnetic fields near their edges. Subsequent work will investigate this effect.

One of the great strengths of this system also turned out to be a disadvantage. Because the field lines along each axis were much straighter than in a single-coil system, tags rotated parallel to one axis and perpendicular to the other two were undetectable by the coils along the perpendicular axes. This proves to be a problem for determining the three-dimensional location of a tag, since some signal needs to be seen by all six coils to precisely find the location and orientation.

The simple solution to this is to use a three-tag object, similar to the cube and eyeball in the Musical Trinkets, in order to ensure that a signal is received for a given object on all three axes. To conserve frequency usage it may be possible to use three tags with identical resonant frequencies, as they would be representing the same object. Likewise, orthogonal coils could be driven together to produce curved field geometries that could resolve ambiguities (with significantly more math). For the purposes of using only one tag, though, it would also be possible to increase the sensitivity of the system and assume no change in the position of a tag along one axis when its signal drops out on that axis, updating the position once the signal is reacquired. Under normal usage it seems unlikely that a tag will spend any significant length of time completely undetectable along any given axis.

Using a three-tag object, several series of measurements were taken for the system. Holding the tags strictly parallel to their corresponding coil axes, the strength of the signal for each tag from each coil along a 5 x 5 x 5 grid was recorded, normalized, and compared. Because of the relatively high baselines, which caused tags not to be detected at all by some coils, we simply took the differences between normalized signals, instead of dividing the difference by the sum. Normalization involved dividing each signal by the highest value recorded for that coil/tag combination at any position.

The differences between corresponding signals is plotted on a three-dimensional graph in Figure 28, which shows that the raw information from the system is already not too bad for three-dimensional position extrapolation. Magnetic field leakage, caused by spacing the coils farther apart than the optimal Helmholtz distance, causes some warping of the field near the edges; this is most apparent in the lowest plane plotted in the figure, which happened to be slightly outside of the perimeter of the cube. However, because we receive plenty of information from the tags and the positional data is uniform and not multiply-valued, a simple fit algorithm could be designed to compensate for this and allow for improved accuracy in position determination.

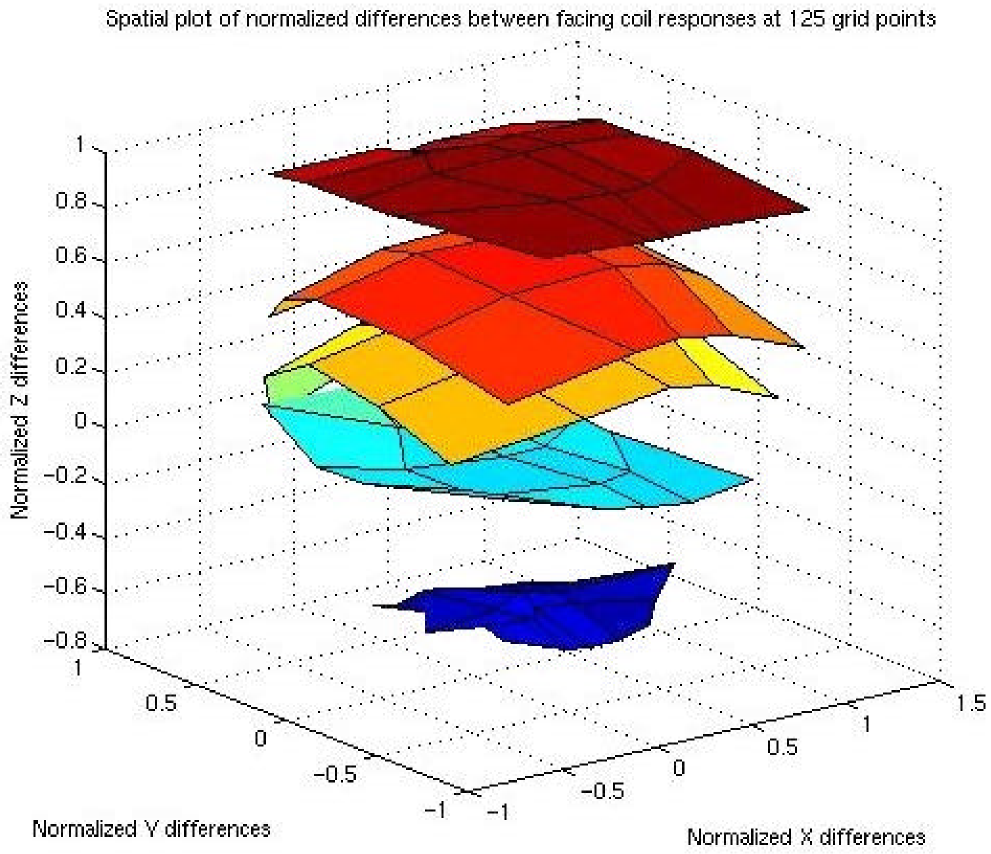


Figure 28. Mesh visualization of raw data from tag readers along an evenly-spaced 5x5x5 grid. Although results were irregular, relative positions were fairly conserved and so a simple fit algorithm could probably provide decent accuracy.

Noise data was also taken for three points within the cube. In the center of the cube, the returned signals for a tag being held in place varied by up to 8% on average for the noisiest coil, and towards the edge of the cube, the signals varied by up to 6% on average for the noisiest relevant coil, where relevance was determined by which coil each tag was on-axis with. This noise level, while not as low as in the single-coil version, is sufficiently low to ensure that the precision for the system would be acceptably good for simple interface applications given a properly-written fit algorithm.

Finally, a tag in the middle of the cube was rotated about the three axes to demonstrate the correlation between the signals coming from the three sets of coils and the orientation of the tag. These signals are plotted against time in Figure 29. The changes in the response along one axis for one tag is accompanied by corresponding changes in the

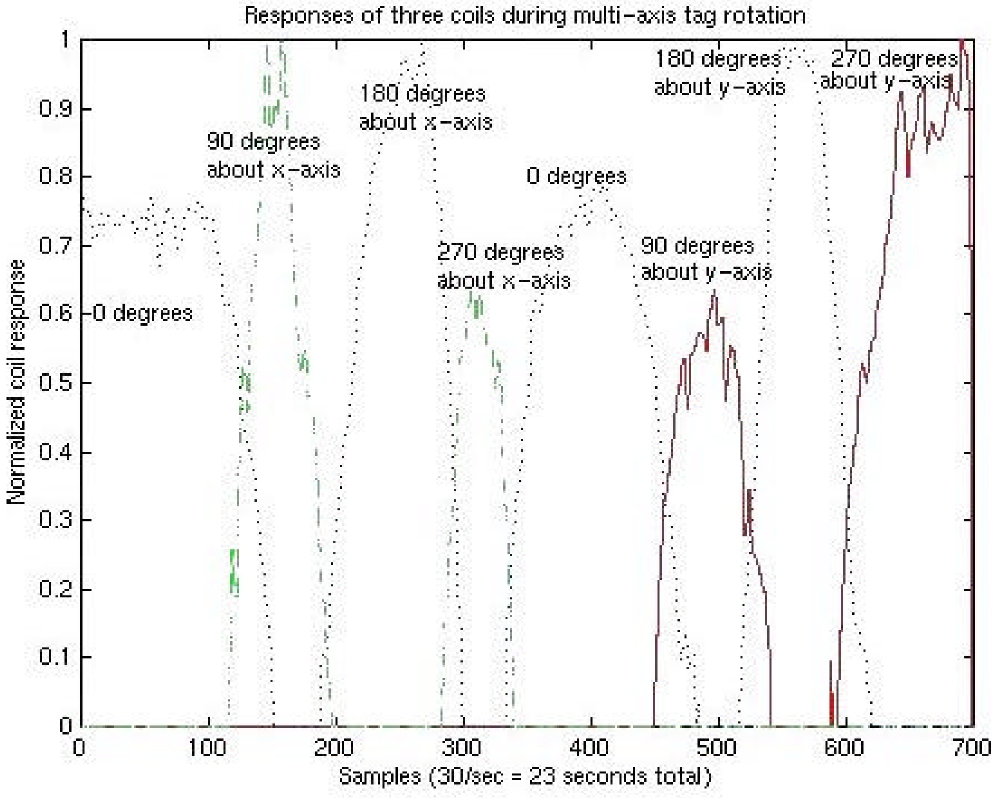


Figure 29. Normalized responses from three coils for one tag during tag rotation around coil axes. As the tag signal decreases along one axis, it increases along another. Labels indicate rotation represented by each graph peak.

other axes, making it likely that a good fit algorithm would be able to decouple the axes without too much difficulty.

A simple graphical application was written to demonstrate the use of the system with one tag, which, despite the sensitivity issues with perpendicular axes mentioned above, produced a usable result. The application drew a dot on the screen when a tag was detected, which moved around the screen based on the tag's position in two axes and became larger and smaller based on the tag's position in the third axis. Coordinates were determined simply by difference-over-sum, as described earlier. Furthermore, the disc changed color depending on the orientation of the tag, with each axis of orientation controlling one primary color component (red, green, blue).

This application successfully showed the basic feasibility of the six-coil tracking system prototype. Other improvements to the system are left for future work.

5.6. Six-coil tag reader conclusions

Of the methods we envisioned as possibilities for extending the detection abilities of the single-coil swept-frequency tag reader, we selected a concept based upon having six coils wound around the faces of a cubical frame. This method was selected for its potential to determine position in three dimensions, which in turn was motivated by our hopes to extract more information for each tag, as well as the apparent expectations of people using the one-coil system. Other advantages of using this configuration included its field-shaping ability; because each pair of coils could shape field lines such that the area in between the coils had straighter field lines than the area surrounding a single lone coil, tags would produce more uniform signals based on their position along the coils' axis regardless of their position perpendicular to the axis.

By using multiplexers to alternate signals between the three pairs of coils, data from each of the three dimensions could be gathered at the expense of a threefold decrease in detection speed. Precision and accuracy of the data acquired were sufficient for applications that we anticipate, and with the addition of an algorithm to fit the irregularities of the data to the three-dimensional space, we expect that the results would allow a great deal of control in three dimensions. The raw data already has enough regularity to be useful in itself, and supports the potential for accurate 3-axis tracking after calibration.

Several immediate improvements could be made to the system at this point. First, the fit algorithm could be added to the host computer, in order to map the raw data coming from the coils to the actual position of the tags. Also, the baseline of each reader could be taken at multiple points, instead of using a static baseline to determine presence of a tag in the field. This would improve precision, as signals would be detected at a lower threshold. Furthermore, the noise level of this system is higher than in the single-coil system, and investigation into the nature of the noise, some of which seemed likely to have been caused by loose connections and some of which seemed to arise from unexpected interactions between reader boards or coils, would improve the precision substantially. Finally, with appropriate field-shaping and design, the size of the cube could be expanded, which would make the frame less of an obstruction to user interaction.

As it stands, the six-coil tag reader could already be useful in a number of applications. Having a system that detects the motion of objects in real-time within a three-dimensional space makes possible straightforward interactions, such as moving an object within the field and controlling an onscreen counterpart. Given the ability of this system to handle fairly large numbers of tags, they could be placed on all the fingers of a hand, and tracked independently, creating an effect similar to a virtual reality glove, except without wires, or an actual glove, or providing power to devices attached to the hand itself. Similarly, the six-coil tag reader offers advantages over vision-based or sonar-based interaction systems, in that multiple objects or fingers will not occlude each other so that real-time interaction can continue uninterrupted regardless of position. Compared to motion-capture technology like those offered by companies such as Polhemus, our system might provide a simpler alternative for use with a smaller region.

In the next chapter we will discuss other possible uses for future versions of this technology as well.

Chapter 6

Conclusion

In this thesis we explored the design and development of passive tag readers for application in tangible-object desktop interfaces. Such interfaces would allow intuitive interactions between a computer and a user not currently enabled by typical input devices. Some of the capabilities that we deemed important in such a system were the ability to handle multiple tags, the ability to sense continuous positions of the tags in real-time, and an acceptable desktop read range, with reasonable stability, adjustability, precision, and accuracy.

6.1. The ringdown tag reader

The first tag reader design we examined was the ringdown reader. The basic concept behind this involved a coil through which we transmitted pings at the resonant frequency of a predetermined tag, and then the system was made quiet in order to listen for the ringdown responses of the tag. In order to detect multiple tags, we cycled our pings between the resonant frequencies of multiple tags. For real-time, continuous interaction, we performed the transmit-receive cycle in 1/30 seconds and used a gated integrator to sum the strength of the demodulated received signal.

By adding a ladder of MOSFET's and triacs, we were able to dynamically retune the coil in order to optimize its resonance to match each successive tag being detected. This enabled higher transmit power and better receive sensitivity. Although another design possibility was to transmit broadband noise to excite all tags with a single ping, this had implications limiting the transmit power. We used magnetostrictor tags for their high Q values, hence strong response.

The system produced a usable result with the desired real-time and continuous performance on multiple tags. Detection of one tag took about 12 milliseconds, which gave essentially real-time performance with three tags; however, beyond about eight tags, the speed of the interaction would suffer because each tag requires incrementally more time for ringdown and detection. The other primary difficulty with the ringdown system was its need to be preprogrammed for each resonant tag and the resulting vulnerability to drift. Nonetheless, the ringdown reader has good power output and very capably detects a small number of tags, making it a good choice for simple applications.

6.2. The swept-frequency tag reader

Next, we examined another approach, that of using a transmitted signal sweeping continuously through the relevant frequency range. Because a continuous sweep cannot stop to listen for ringdown responses, our reader used an inductive bridge circuit to detect the energy being drawn dynamically from the magnetic field of the reader coil by a resonant tag.

We decided to use LC tags for the ease of adjusting their frequencies. Our swept-frequency reader swept a range of frequencies spanning the resonances of all tags. A reverse exponential sweep ensured that lower frequencies received proportionally more time for tag excitation. By running the sweep thirty times a second, we were able to produce essentially real-time interaction, and by synchronously demodulating the output of the differential amplifier of the inductive bridge, we were able to determine the magnitude of a signal coming from a tag in the field.

Our finished reader had all the expected properties; it could sweep thirty times a second and detect the continuous signal strength from all the tags in the frequency range. Its receive range was slightly over one foot, which is acceptable for desktop-scale interaction, and its jitter level was remarkably quiet. Frequency drift was not a severe drawback except under relatively large temperature changes; even then, tag selection allowed for some level of tolerance, and the addition of an algorithm to monitor and compensate for frequency drift by comparing frequencies against the microprocessor's clock would remove this problem altogether. In many respects, the swept-frequency tag reader made an ideal tangible interface backend.

6.3. Swept-frequency tag reader applications

The earliest swept-frequency tag reader was demonstrated using tags embedded within Lego bricks, and produced simple tones when it detected the tags. Additional tag interactions were produced by placing three orthogonal tags in a single object, thus allowing for determination of orientation, as well as by perturbing the resonance of another tag using a slider, thus allowing for an additional degree of control. These basic interaction concepts have been used in subsequent applications as well.

Hiroshi Ishii's Tangible Media Group also used the swept-frequency tag reader as a backend for their musicalBottles project, which played various lines of music when bottles on a tabletop were uncorked. Around the same time, our group began to produce the Musical Trinkets system, which in its final incarnation produced music and graphics on a small countertop with a frosted glass screen mounted on top. Twenty tags embedded in sixteen toy "trinkets" each produced various interactions. Five rings worn on the fingers produced melodic notes and graphical bouncing balls, interposed over a harmony and color changes in the display controlled by three other tags. The remainder of the tags produced various modifications and distortions of the basic music and graphics, including twinkling sounds, vibrato, pitch bend, and several additional harmonizing voices, all accompanied by viewpoint changes, swirling displays, and other additional graphics.

The Musical Trinkets project was displayed at several prominent conferences and was received very well by the attendees as an interesting interface with notable aesthetic qualities. With the success of the single-coil system, we turned our attention to extending the abilities of our tag reader.

6.4. The six-coil swept-frequency tag reader

Using one read coil, it was possible to derive one piece of information about each tag, which happened to be the proximity of the tag to the coil, scaled by the cosine of the angle between the orientation of the tag and the orientation of the field at the tag's location. With some planning and experimentation, we found that using two parallel coils transmitting simultaneously made it possible to deduce both position and orientation of a tag along the axis of the coils, with the added bonus of shaping the magnetic field lines to be more parallel to the axis and uniform in density. Known as Helmholtz coils, we decided to design a fully three-dimensional tag reading system around this principle.

Using a cubical frame with coils wound around each face and six of our swept-frequency tag reader boards, we multiplexed the signals in order to run each pair of facing coils alternately. The uniformity of the interaction produced was encouraging, although careful tuning was necessary to achieve the best results. Using a single object with three orthogonal coils was generally the most reliable way to continuously receive data across all six degrees of kinematic freedom. Data was taken and produced excellent results, even without the addition of a calibration or fitting algorithm, and a simple graphical tag tracking application was written for demonstration purposes.

Our six-coil tag reader is a work in progress, and numerous improvements to the current system remain to be investigated. Beyond that, future improvements can come from redesigns of the entire system and will be discussed in the next section.

6.5. Future directions and applications

All of our systems met basic expectations and demonstrated the feasibility of their underlying concepts for application to tangible interfaces. However, many small improvements can be made to the systems as they stand now, as well as more fundamental redesigns which would enable applications of much greater scope than currently possible.

Both the ringdown tag reader and the swept-frequency tag reader have decent transmit range and receive sensitivity now, but additional work in this area would involve finding higher-quality components and tuning the resonances more carefully in order to optimize the efficiency of the transmit and receive sections, and possibly exploring other alternatives for tag detection beyond inductive bridges. With some effort, both systems could potentially be adapted to much bigger coils and read areas than the 1-2 foot range we have now, with the possibility of encompassing an entire physical desktop area or perhaps even a larger, person-scale space. This would most benefit multiple-coil geometries like the six-coil system, as larger coils would increase the size of the cube and make the frame itself less of an obstruction for desktop interaction.

The swept-frequency tag reader would also benefit from the addition of a multi-point dynamically-sampled baseline for the purposes of determining the threshold above which a tag is detected and reported. The static baseline currently in use is necessarily limited by frequencies in the range with higher noise susceptibility or distorted baselines from dispersive bridge imbalances. Sampling each point of the baseline on system startup would enable tags at less noisy frequencies and at suppressed baselines to be detected at greater range.

Also for the swept-frequency reader, algorithms to compensate for temperature-related frequency drift would be helpful in increasing the stability of the system. The host computer would simply need to receive frequency measurements from the reader board and adjust subsequently reported tag positions accordingly. Note however that tags near the high and low extremes of the frequency range may drift out of range altogether, and compensation algorithms cannot improve this without the addition of a feedback loop directly between the microcomputer and the voltage controlling the oscillator.

Finally, in order to realize the full potential of the six-coil reader, a fit algorithm for the data would allow the precision of the reader to yield excellent accuracy in three dimensions. With the addition of such a fit algorithm, the reader could drive all sorts of tangible interfaces, and the list of possible applications could even grow well beyond what is possible just on a desktop.

On the simple end of tangible interfaces, the common everyday mouse as used today could be adapted to a three-dimensional version, with clicks detectable (and in a continuous manner as well) by tying the mouse "buttons" to changes in tag resonant frequencies. More naturally, using any of the tag readers described, common desktop objects could be moved around and monitored by the computer. In less computer-related environments, for instance, a kitchen countertop with a series of tagged cooking utensils or supplies could observe the location and presence of particular objects and furthermore use resonant frequency distortions to sense the quantity of material present in a container (e.g. as mentioned in Chapter 1).

The previous chapter also observed that the fully three-dimensional nature of the six-coil system enables it to serve as not only a desktop-scale interface with the ability to track an object being moved throughout space, but rings worn on the fingers of a hand could be tracked in real time, enabling an interaction similar to that of virtual-reality gloves. Even with a frame of the size we currently use, a sort of "virtual wetbox" can be generated in this manner, in which the insertion and position of the user's hand within the cubical frame can be reflected in an onscreen representation. As a real wetbox involves fixed gloves between a user and a hazardous manipulable environment in which the user cannot be physically present, this virtual equivalent could provide similar interactions with a computer-generated model.

Furthermore, just as the "wetbox" limits mobility, the cubical frame presents a similar difficulty. However, the optimizations described above create the possibility that the frame can be enlarged to the point that its presence is negligible from a usability standpoint. Even with the frame in place, the ability to track fingers without either the wires or the power sources of current active tracking systems, or the occlusion that happens with current vision-based tracking, has substantial application-dependent advantages, with the frame nicely defining the interaction volume.

Other applications for such a system include motion capture, which is an increasingly important aspect of computer graphics, medicine, and other fields. This has similar requirements (real-time, continuous, etc.) to those of tangible interfaces, except that the data is typically streamed and recorded for later use and analysis instead of immediate interaction. Placing tags on a subject's hand, arm, or other body part that can be immersed within the coil area would allow the tracking of that part in real-time, to be analyzed or converted to other data by a computer. With a larger coil size, a larger part of a body could be encompassed, and could thus provide a simpler alternative to the type of motion-capture currently being offered by companies such as Polhemus,[28] whose predominantly wired active magnetic tracking systems have set a standard for motion capture.

Even further over the horizon, the principles of the six-coil reader, given sufficient refinement, could be used to underlie medical technologies capable of tracking not only body parts, but internal organs as well. This raises the possibility of using small tags, injectable via biopsy needle, onto the surface of an internal organ in order to watch its movement in real time. For instance, cancerous tumors on the surface of organs that move around, such as lungs, force radiation therapy to use relatively diffuse beams for treatment, in order to ensure that the tumor is struck, but this kills much more tissue than necessary.[29] The application of a tracking system such as an improved six-coil reader would enable the beam of radiation to be more tightly focused on the exact tumor site, thus increasing efficiency and decreasing danger to surrounding tissue.

Multiple improvements would have to be made to the multi-coil tag reader technology, but in a relatively controlled RF environment, the needs of increased sensitivity, larger coils, and smaller tags could be attained with further development.

6.6. Closing

We have investigated variations on two types of magnetic tag readers, and our results show that applying these kinds of systems to tangible desktop interfaces can be done cheaply and efficiently, and so common usage of such systems might not be that far off. A plethora of applications is actually made possible by these principles, some as simple and straightforward as three-dimensional computer mice, and others perhaps as far-reaching as large-scale motion capture and cancer therapy.

Our primary hope, though, is simply that as computer technology matures further, interfaces will transit from the direct and literal mouse-and-keyboard paradigm currently in use to encompass a wider variety of interactions, potentially including everyday objects being moved around intuitively. Incorporating these abilities could make computers easier and more natural to use, and may lead to their seamless integration in many aspects of life not currently computerized. The tag readers we described could be instrumental in providing these capabilities, and so we encourage future research in interface development to consider these projects as viable foundations for such work.

References

1. H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms," *Proceedings of the Conference on Human Factors in Computing Systems (CHI '97)*, ACM, Atlanta, March 1997, pp. 234-241.

2. M. Weiser, "The Computer for the 21st Century," *Scientific American*, 265(3), September 1991, pp. 94-104.

3. K. Finkenzeller, *RFID Handbook*, John Wiley & Son, Ltd., New York (1998), p. 26.

4. "BiStatix Technology: A Whitepaper", available at http://www.motorola.com/GSS/SSTG/smartcard/white_papers/BiStatix_Whitepaper.pdf

5. See http://www.sensormatic.com/eas_www/easum.htm

6. K. Finkenzeller, *RFID Handbook*, John Wiley & Son, Ltd., New York (1998), p. 32.

7. K. Finkenzeller, *RFID Handbook*, John Wiley & Son, Ltd., New York (1998), p. 30.

8. R. Fletcher, J. A. Levitan, J. Rosenberg, and N. Gershenfeld, "Application of Smart Materials to Wireless ID Tags and Remote Sensors," *Materials for Smart Systems II, Proceedings of the 1997 MRS Symposium*, Boston, MA, Materials Research Society, Pittsburgh, PA (1997), pp. 557-562.

9. W. B. Spillman, Jr., S. Durkee, and W. W. Kuhns, "Remotely Interrogated Sensor Electronics (RISE) for Smart Structures Applications," *Proceedings of the SPIE Second European Conference on Smart Structures and Materials*, Volume 2361, Glasgow (October, 1994), pp. 282-284.

10. J.A. Corbyn, "Pulse Induction Metal Detector," *Wireless World*, March 1980, pp. 40-44.

11. E. Fukushima, S.B.W. Roeder, *Experimental Pulse NMR: A Nuts and Bots Approach*. Addison-Wesley Publishing Company, New York (1981).

12. Hap Patterson, Sensormatic Corporation, personal communication, Spring 2000.

13. D. Lancaster, *Active Filter Cookbook,* Synergetics Press, Arizona (1995), p. 200.

14. Hap Patterson, Sensormatic Corporation, personal communication, Spring 2000.

15. R. Fletcher, O. Omojola, E. Boyden, and N. Gershenfeld, "Reconfigurable Agile Tag Reader Technologies for Combined EAS and RFID Capability," *Proceedings of the 2nd IEEE Workshop on Automatic Identification Advanced Technologies*, Summit, New Jersey (1999).

16. K. Finkenzeller, *RFID Handbook*. John Wiley & Sons Ltd., England (1999), p. 27.

17. W.G. Jung, *IC Timer Cookbook*, Howard W. Sams & Co., Inc., Indianapolis (1977), p. 107.

18. P. Horowitz and W. Hill, *The Art of Electronics*, Cambridge University Press, New York (1989), p. 992.

19. J. Paradiso and K. Hsiao, "Swept Frequency, Magnetically-Coupled Resonant Tags for Realtime, Continuous, Multiparameter Control," *Human Factors in Computer Systems; CHI99 Extended Abstracts,* ACM Press, New York (1999), pp. 212-213.

20. K. Hsiao and J. Paradiso, "A New Continuous Multimodal Musical Controller Using Wireless Magnetic Tags," *Proceedings of the 1999 International Computer Music Conference*, International Computer Music Association, Sna Francisco (October 1999), pp. 24-27.

21. J. Paradiso, K. Hsiao, J. Strickon, J. Lifton, A. Adler, "Sensor systems for interactive surfaces," *IBM Systems Journal*, Vol. 39 No. 4 (2000), pp. 892-914.

22. H. Ishii, R. Fletcher, J. Lee, S. Choo, J. Berzowska, C. Wisneski, C. Cano, A. Hernandez, C. Bulthaup, "musicBottles," *SIGGRAPH 1999 Conference Abstracts and Applications*, ACM Press, New York (1999).

23. J. Paradiso, K. Hsiao, A. Benbasat, "Musical Trinkets: New Pieces to Play," *SIGGRAPH 2000 Conference Abstracts and Applications*, ACM Press, NY, July 2000, p. 90.

24. M. Downie, *Behavior, Animation and Music: The Music and Movement of Synthetic Characters*. M.S. Thesis, MIT Media Lab (January 2001).

25. See http://www.smau.it

26. J. Smith, *Electric Field Imaging*. Ph.D. thesis, MIT Media Lab (February 1999), page 49.

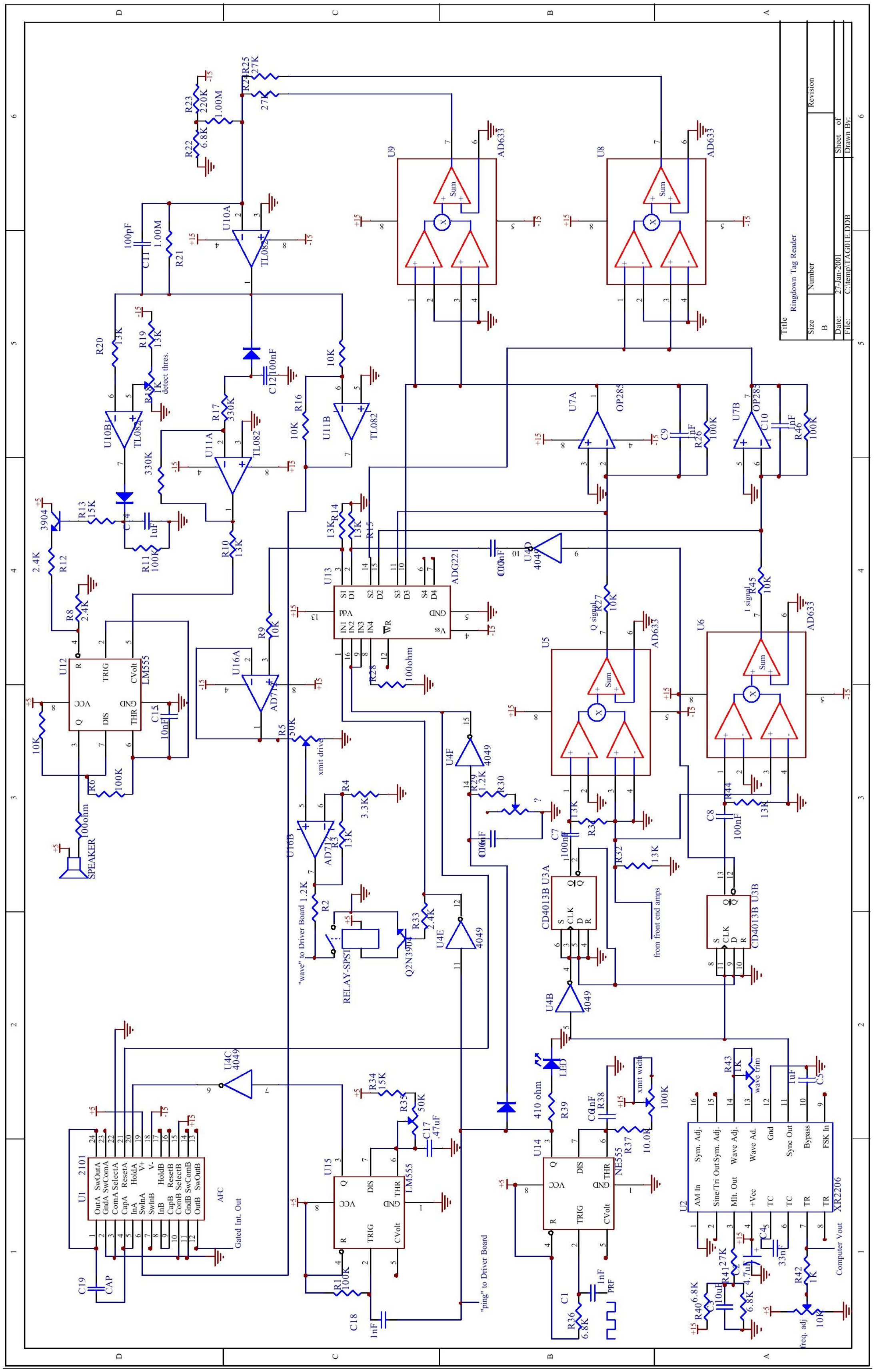
27. J.R. Reitz, F.J. Milford, *Foundations of Electromagnetic Theory*, Addison-Wesley Publishing Company, Reading, Massachusetts (1967), p. 156.

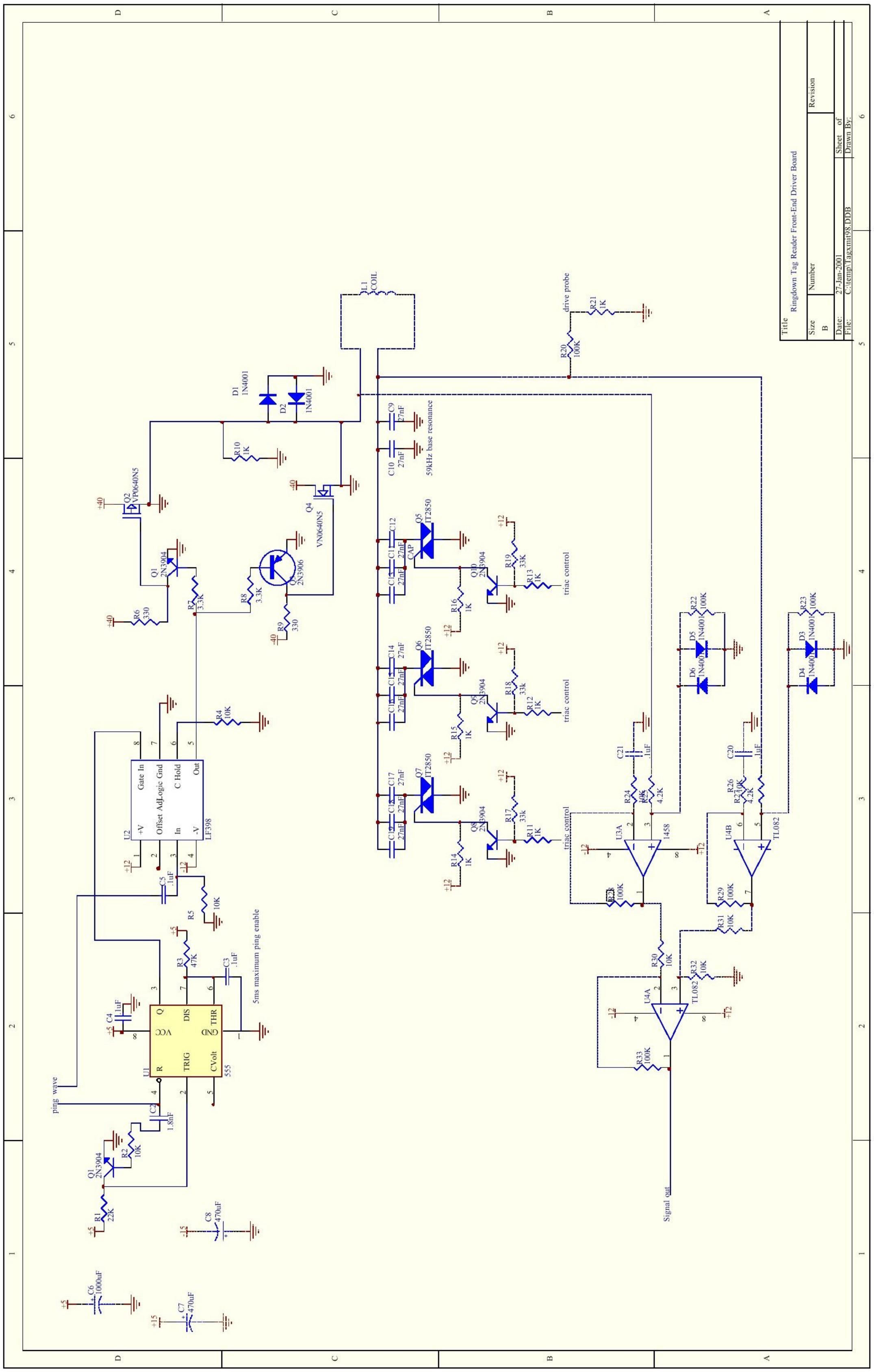
28. See http://www.polhemus.com

29. Dr. Paul G. Seiler, Paul Scherrer Institut (PSI), Villigen, Switzerland, personal correspondence, Spring 2000.

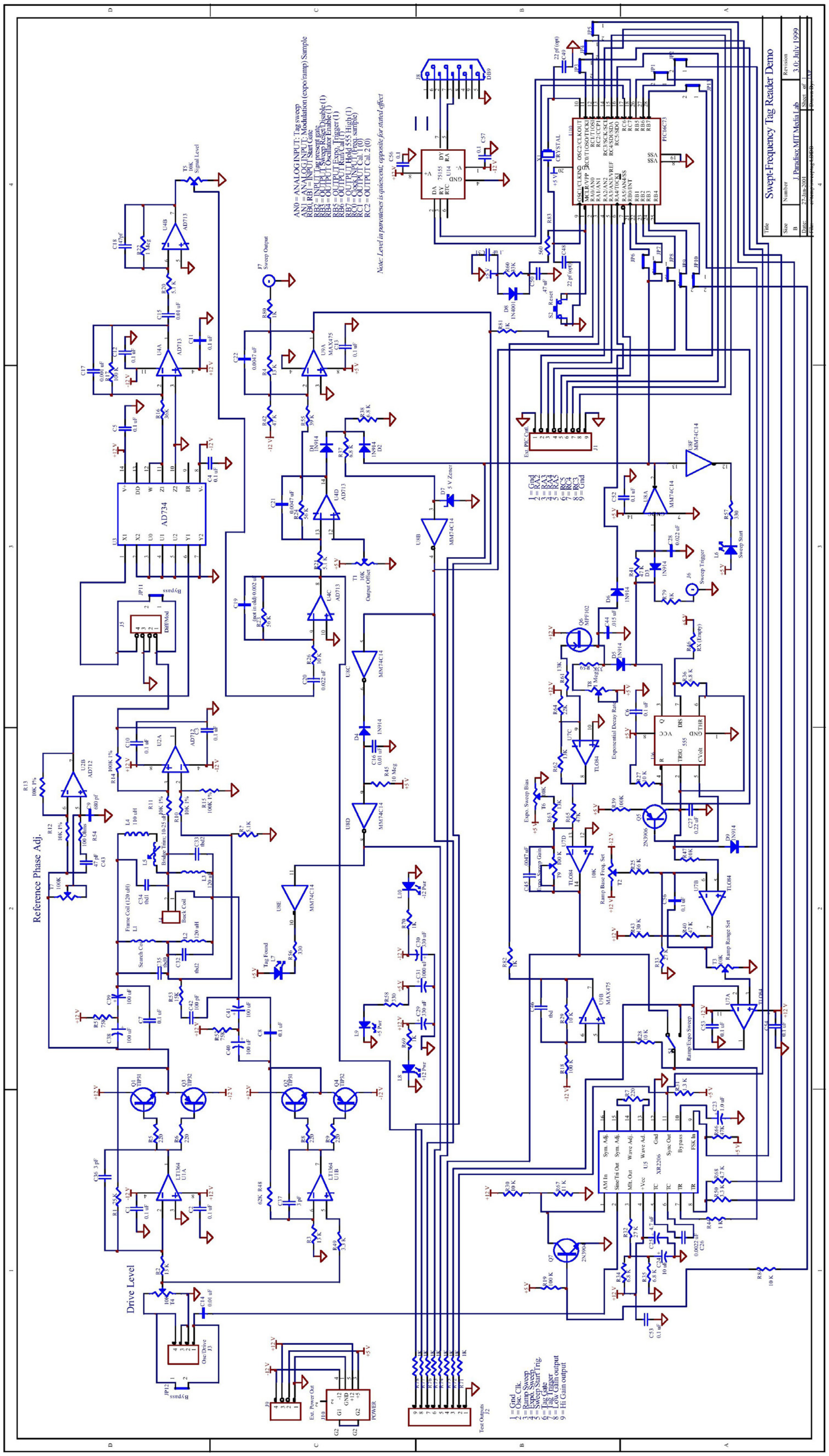
Appendix A

This appendix presents the schematics for the ringdown tag reader, the ringdown tag reader front end driver, and the swept-frequency tag reader.


Title
Ringdown Tag Reader
Size
B
Number
Revision
Date: 27-Jan-2001
Sheet of
File: C:\temp\TAG01E.DDB
Drawn By:
U9 AD633
U8 AD633
Sum
U10A TL082
U10B TL082
U11A TL082
U11B TL082
R25 27K
R24 27K
R23 220K
R22 6.8K
1.00M
C11 100pF
R21 1.00M
R20 13K
R19 13K
R18 1K
detect thres.
C12 100nF
R17 330K
330K
R16 10K
10K
R13 15K
3904
C14 1uF
R11 100K
R10 13K
R12 2.4K
R8 2.4K
U12 LM555
C15 10nF
R6 100K
10K
100ohm
SPEAKER
U13 ADG221
R14 13K
R15 13K
C13 10nF
U4D 4049
U7A OP285
C9 1nF
R26 100K
U7B OP285
C10 1nF
R46 100K
R27 10K
Q signal
R45 10K
I signal
U5 AD633
U6 AD633
R28 100ohm
U16A AD712
R9 10K
R5 50K
xmit drive
R4 3.3K
U16B AD712
R3 13K
R2 1.2K
"wave" to Driver Board
RELAY-SPST
Q2N3904
R33 2.4K
U4E 4049
U4F 4049
R29 1.2K
R30
?
C6 100nF
C7 100nF
R31 13K
R32 13K
from front end amps
CD4013B U3A
CD4013B U3B
C8 100nF
R44 13K
U4B 4049
LED
R39 410 ohm
U14 NE555
C38 10nF
R37 10.0K
xmit width
100K
C1 1nF
PRF
R36 6.8K
U4C 4049
U15 LM555
R34 15K
R35 50K
C17 .47uF
R1 100K
C18 1nF
"ping" to Driver Board
U1 2101
C19 CAP
AFC
Gated Int. Out
U2 XR2206
R43 1K
wave trim
C5 1uF
C4 33nF
R41 27K
C3 4.7uF
C2 10uF
R40 6.8K
R38 6.8K
R42 1K
Computer Vout
freq. adj
10K



Title
Ringdown Tag Reader Front-End Driver Board
Size
B
Number
Revision
Date: 27-Jan-2001
Sheet of
File: C:\temp\Tagxmit98.DDB
Drawn By:
L1 COIL
D1 1N4001
D2 1N4001
R10 1K
Q2 VP0640N5
Q4 VN0640N5
Q3 2N3906
Q1 2N3904
R7 3.3K
R8 3.3K
R6 330
R9 330
+40
C9 27nF
C10 27nF
59kHz base resonance
C12
C11 27nF
CAP
Q5 IT2850
Q10 2N3904
R19 33K
R13 1K
R16 1K
triac control
Q6 IT2850
C14 27nF
Q9 2N3904
R18 33k
R12 1K
R15 1K
triac control
Q7 IT2850
C17 27nF
Q8 2N3904
R17 33k
R11 1K
R14 1K
triac control
+12
drive probe
R20 100K
R21 1K
R22 100K
D5 1N4001
D6 1N4001
R23 100K
D3 1N4001
D4 1N4001
C21 .1uF
C20 .1uF
R24 10K
R25 4.2K
R26 10K
R27 4.2K
U3A 1458
U4B TL082
R28 100K
R29 100K
R31 10K
R30 10K
R32 10K
U4A TL082
R33 100K
-12
Signal out
U2 LF398
+V
Offset Adj
Logic Gnd
Gate In
In
C Hold
-V
Out
R4 10K
C5 .1uF
R5 10K
U1 555
R
VCC
Q
DIS
THR
GND
TRIG
CVolt
R3 47K
C3 .1uF
5ms maximum ping enable
C4 .1uF
ping wave
C2 1.8nF
R2 10K
Q1 2N3904
R1 22K
+5
C8 470uF
C6 1000uF
C7 470uF
+15
D
C
B
A
1
2
3
4
5
6



Swept-Frequency Tag Reader Demo
J. Paradiso, MIT Media Lab
3.0, July 1999
Note: Level in parentheses is quiescent, opposite for stated effect
Drive Level
Reference Phase Adj.
AD734
PIC16C73
XR2206
555
Test Outputs
POWER
Ext. Power Out
Sweep Output
Signal Level
Sweep Trigger
Sweep Start
Tag Found
Exponential Decay Rate
Ramp Range Set
Ramp Base Freq. Set
Ramp/Expo Sweep
Expo. Sweep Bias
1 = Gnd
2 = Osc. Clk
3 = Ramp Sweep
4 = Expo. Sweep
5 = Sweep Start Trig.
6 = Tag Gate
7 = Tag Trigger
8 = Low Gain output
9 = Hi Gain output

Appendix B

Here we present the information necessary to implement the frequency drift compensation algorithms described in section 3.5.

On each normal sweep, the tag reader samples the output voltage of the tag system several hundred times, and for each peak that exceeds the baseline threshold, it outputs relevant information. For one sweep, the serial transmission provides the following information:

1. The start time (where "time" is the counter of samples taken so far this sweep) of each peak,
2. The stop time of each peak,
3. The sum of all samples under the peak (essentially an integral),
4. The number of samples taken in the course of the entire sweep.

By using frequency information and the number of samples to adjust the measured start time and stop time, we can compensate for frequency drift.

On each frequency calibration sweep, the tag reader outputs samples of the frequency at periodic intervals during the sweep. In a typical sweep, ten samples are returned. When more samples are returned we can ignore the samples beyond the tenth; although these are part of the sweep, tags detected beyond the tenth sample may not be picked up under all common temperature conditions (one of the timing issues discovered while taking heat measurements for section 3.5).

First, let us derive the compensation algorithm for the linear sweep. Let t_{start} be the start time for a tag, t_{stop} be the stop time of the tag, and n be the number of samples taken during an initial calibration sweep for that tag. Further, let us take frequency samples so that f_1 and f_{10} are the first and tenth frequency samples in that initial calibration sweep. Then, our host computer's program can store a value for the start and stop frequencies of the tag that has been converted from the arbitrary counter timing of the PIC code into something more closely resembling the actual frequencies of the tag.

This can be done by scaling each time *t* such that $f_{real} = \frac{t}{n}(f_1 - f_{10}) + f_{10}$. After this, our host computer can perform its computations by identifying tags based on their true frequencies. Then, as time passes, temperatures change, and the frequency range drifts, we can take new frequency samples periodically. Based on the new frequency measurements, each subsequent incoming tag start and end time can be scaled also by the equation just given, and tag identification can proceed based on the new scaled frequencies, using the same lookup table as the one generated on startup.

The compensation algorithm for the exponential sweep works exactly the same way, except that three frequencies are necessary to scale the tag time into an actual frequency. By using f_1, f_5, and f_9, the first, fifth, and ninth frequency samples, we can scale a time given by the PIC into a real frequency once we can derive the exponential function which would produce f_1, f_5, and f_9. We use these particular values because they are spaced equidistantly in time, which simplifies the math. By scaling the time axis so f_1 occurs at time 0, f_5 occurs at time 1, and f_9 occurs at time 2, we find we have to solve for A, B, and k in a system of three equations

$$Ae^{-0k} + B = f_1$$

$$Ae^{-k} + B = f_5$$

$$Ae^{-2k} + B = f_9$$

Taking logarithms of the second and third equation, it becomes possible to scale both equations to be equal to *k*. Removing *k* and then substituting *B* using the first equation, we obtain

$$A = \frac{(f_5 - f_1)^2}{2(f_9 - f_5)}$$

Thanks to the first equation, we thus know that

$$B = f_1 - A$$

and finally we have

$$k = -\ln\frac{f_5 - B}{A}$$

Given the values of A, B, and k, we can derive a number analogous to the actual frequency for each reported tag start and stop time t using the relation

$$f_{real} = Ae^{-20kt/9n} + B$$

As with the linear case, by recording the values of f_{real} for each t reported by the tag reader, we can then use these values to distinguish between tags by their frequency in the host computer, and when frequency measurements appear in periodic calibration updates, it is possible to adjust the coefficients accordingly using the above relations, which then continue to allow subsequent values to be scaled.

With a system like this implemented, it should then be possible to run the swept-frequency tag reader for extended periods of time without the need to recalibrate for the frequency drift of the reader. Frequency drift in the tags themselves is much harder to compensate, but apart from extreme heat and physical distortion, the tags tend to be more stable than the reader, particularly because the tags are in open air at room temperature, while the reader board is often near other equipment (e.g. video projectors) which alters its ambient temperature more often.

Appendix C

In this appendix we present the code used in the various systems described in this thesis. The final PIC used with the PIC code was the Microchip PIC16F873. C code was compiled using CCS PIC C (see http://www.ccsinfo.com) and written using MPLAB (see http://www.microchip.com) using a PICSTART Plus chip programmer (also available from Microchip).

Windows C++ code was written and compiled in Microsoft Visual Studio, with the addition of Rogus McBogus, a MIDI library developed by the Brain Opera group at the Media Lab, and with GLUT, the extension library for OpenGL (information available at http://reality.sgi.com/opengl/glut3/glut3.html). The code used for the Musical Trinkets is very long and will not be included here. Links and information are current as of January 2001; contact the author for further information.

C.1. Swept-Frequency Master Board PIC

The first program is the one used to control the PIC on the master tag reader board in the six-coil tag system, which can be used as is to control a single tag reader as well.

```
#include "tag10pic.h"

#define PIN_TRIGGER PIN_B1
#define PIN_CAL 1
#define PIN_INPUT 0
#define PIN_EXPORESET PIN_B0
#define PIN_TAGPRESENT PIN_B2
#define PIN_RESETSWEEP PIN_B3
#define PIN_OSCILLATE PIN_B4
#define PIN_EXPOTRIG PIN_B5
#define PIN_RUNCAL PIN_B6
#define PIN_HI555 PIN_B7
#define PIN_GATE PIN_C1
#define PIN_MUX0 PIN_C4
```

```
#define PIN_MUX1 PIN_C5

#fuses HS,NOWDT,NOPROTECT,PUT,NOBROWNOUT
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)
#use fast_io(B)
#use fast_io(C)

typedef union {
  unsigned long two;
  unsigned int ones[2];
} WORD;

unsigned int i;
WORD tmp;
unsigned int overflows=0;
unsigned long threshold=0;
unsigned int counter=0;
unsigned int last, start;
char axis=0; // which pair of boards are we running now?  control the mux
int running=1;
char sweeping=1;
WORD sum;
unsigned int bufstart=0, bufend=0, buffer[80];

void addbuf(unsigned int val)
{
  if (bufend==bufstart-1 || (bufend==79 && bufstart==0)) return;
  buffer[bufend++]=val;
  if (bufend>79) bufend=0;
}

unsigned int getbuf()
{
   unsigned int val;
   if (bufend==bufstart) return 255;
   val=buffer[bufstart++];
   if (bufstart>79) bufstart=0;
   return val;
}

main()
{
#asm
bsf 0x9f, 7
#endasm
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)
   setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
   setup_port_a(ALL_ANALOG);
   setup_adc(ADC_CLOCK_INTERNAL);
   set_tris_b(0b00000111);
   set_tris_c(0b10000000);
   printf("PIC starting up...\n");
   output_high(PIN_RESETSWEEP);
   output_high(PIN_OSCILLATE);
   output_high(PIN_EXPOTRIG);
   output_high(PIN_RUNCAL);
   output_high(PIN_HI555);
   output_low(PIN_MUX0);
   output_low(PIN_MUX1);
   while (!input(PIN_TRIGGER));
   while (input(PIN_TRIGGER));
   for (i=0; i<127; i++)
   {
      set_adc_channel(PIN_INPUT);
      delay_us(45);
      while (!input(PIN_TRIGGER))
      {
         tmp.two=read_adc();
         if (tmp.two>threshold)
            threshold=tmp.two;
      }
      printf("%ld ", threshold);

      while (input(PIN_TRIGGER));
   }


  setup_counters(RTCC_INTERNAL, RTCC_DIV_4);
  enable_interrupts(RTCC_ZERO);
  enable_interrupts(GLOBAL);

   while(1)
   {
      set_adc_channel(PIN_INPUT);
      while (!input(PIN_TRIGGER));
      disable_interrupts(GLOBAL);
      addbuf(overflows);
      addbuf(counter);
      addbuf(255);
      addbuf(255-axis);
      counter=0;
      overflows=0;

      axis++;
      if (axis>2) axis=0;
      if (axis==1) output_high(PIN_MUX0); else output_low(PIN_MUX0);
      if (axis==2) output_high(PIN_MUX1); else output_low(PIN_MUX1);
while (input(PIN_TRIGGER))
      {
         if (bufstart!=bufend)
         {
            printf("%c", getbuf());
         }
enable_interrupts(GLOBAL);

   }
}

#INT_RTCC
void loop()
{
         int start_o;
         set_rtcc(100);


         if (counter==255) {overflows++; counter=0;}
         else counter++;

         tmp.two=read_adc();
         if (tmp.two>threshold)
         {
             if (last==0)
             {
                start=counter;
//                start_o=overflows;
             }
             last=1;
             sum.two+=(tmp.two);
         }
         else
         {
            if (last==1)
            {
//               addbuf(start_o);
               addbuf(start);
//               addbuf(overflows);
               addbuf(counter);
               addbuf(sum.ones[1]);
               addbuf(sum.ones[0]);
               last=0;
               sum.two=0;
            }
         }

   if ((!(counter&0b00000011)) && (bufstart!=bufend))              // every 4 counts
   {
      printf("%c", getbuf());
   }

}
```

C.2. Swept-Frequency Frequency-Drift Collector

This next program is similar to the first but its sole purpose is to return frequency and voltage samples from the voltage-controller oscillator. Added to the code in C.1 and combined with the algorithms for the host computer given in Appendix B, automatic frequency drift compensation should be easily implementable.

```
#include "tag10cal.h"

#define PIN_TRIGGER PIN_B1
#define PIN_CAL 1
#define PIN_INPUT 0
#define PIN_EXPORESET PIN_B0
#define PIN_TAGPRESENT PIN_B2
#define PIN_RESETSWEEP PIN_B3
#define PIN_OSCILLATE PIN_B4
#define PIN_EXPOTRIG PIN_B5
#define PIN_RUNCAL PIN_B6
#define PIN_HI555 PIN_B7
#define PIN_GATE PIN_C3
#define PIN_MUX0 PIN_C4
#define PIN_MUX1 PIN_C5

#fuses HS,NOWDT,NOPROTECT,PUT,NOBROWNOUT
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)
#use fast_io(B)
#use fast_io(C)

typedef union {
  unsigned long two;
  unsigned int ones[2];
} WORD;

unsigned int i;
WORD tmp;
unsigned int overflows=0;
unsigned long threshold=0;
unsigned int counter=0;
unsigned int last, start;
char axis=0; // which pair of boards are we running now?  control the mux
int running=1;
char sweeping=1;
WORD sum;

void exp_recal()
{
  WORD vtmp1, vtmp2, vstmp1, vstmp2, vstmp3;
  WORD ftmp1, ftmp2, ftmp3;

  // get voltage and frequency 1st time
while (1) {
  while(!input(PIN_TRIGGER))
    ;
  while(input(PIN_TRIGGER))
    ;
  while(!input(PIN_TRIGGER))
    ;
  while(input(PIN_TRIGGER))
    ;
  while(!input(PIN_TRIGGER)) {

    vstmp1.two = read_adc();
    set_timer1(0);
    delay_ms(1);
    ftmp1.two=get_timer1();

    if (input(PIN_TRIGGER)) break; // abort if cycle ended

    printf("%c", vstmp1.ones[1]);
    printf("%c", vstmp1.ones[0]);
    printf("%c", ftmp1.ones[1]);
    printf("%c", ftmp1.ones[0]);
  }
  printf("%c%c", 255, 255);

}
}

main()
{
#asm
bsf 0x9f, 7
#endasm
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)
   setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
   setup_port_a(ALL_ANALOG);
   setup_adc(ADC_CLOCK_INTERNAL);
   set_tris_b(0b00000110);
   set_tris_c(0b10000011);
   printf("PIC starting up...\n");
   output_high(PIN_RESETSWEEP);
   output_high(PIN_OSCILLATE);
   output_high(PIN_EXPOTRIG);
   output_high(PIN_RUNCAL);
   output_high(PIN_HI555);
   output_low(PIN_MUX0);
   output_low(PIN_MUX1);
  set_adc_channel(PIN_CAL);

  setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
  exp_recal();
}
```

C.3. PIC Header File

This header file is the one included in the programs in both C.1 and C.2.

```
//////// Standard Header file for the PIC16F873 device ////////
#device PIC16F873
#use delay(clock=20000000)
#nolist
////////////////////////////// I/O definitions for INPUT() and OUTPUT_xxx()
#define PIN_A0  40
#define PIN_A1  41
#define PIN_A2  42
#define PIN_A3  43
#define PIN_A4  44
#define PIN_A5  45

#define PIN_B0  48
#define PIN_B1  49
#define PIN_B2  50
#define PIN_B3  51
#define PIN_B4  52
#define PIN_B5  53
#define PIN_B6  54
#define PIN_B7  55

#define PIN_C0  56
#define PIN_C1  57
#define PIN_C2  58
#define PIN_C3  59
#define PIN_C4  60
#define PIN_C5  61
#define PIN_C6  62
#define PIN_C7  63

////////////////////////////// Useful defines
#define FALSE 0
#define TRUE 1

#define BYTE int
#define BOOLEAN short int

#define getc getch
#define getchar getch
#define puts(s) {printf(s); putchar(13); putchar(10);}
#define putc putchar

////////////////////////////// Constants used for RESTART_CAUSE()
#define WDT_FROM_SLEEP  0
#define WDT_TIMEOUT     8
#define MCLR_FROM_SLEEP 16
#define NORMAL_POWER_UP 24
////////////////////////////// Constants used for SETUP_COUNTERS()
#define RTCC_INTERNAL   0
#define RTCC_EXT_L_TO_H 32
```

```
#define RTCC_EXT_H_TO_L 48
#define RTCC_DIV_2       0
#define RTCC_DIV_4       1
#define RTCC_DIV_8       2
#define RTCC_DIV_16      3
#define RTCC_DIV_32      4
#define RTCC_DIV_64      5
#define RTCC_DIV_128     6
#define RTCC_DIV_256     7
#define WDT_18MS         8
#define WDT_36MS         9
#define WDT_72MS        10
#define WDT_144MS       11
#define WDT_288MS       12
#define WDT_576MS       13
#define WDT_1152MS      14
#define WDT_2304MS      15
#define L_TO_H              0x40
#define H_TO_L                 0

#define RTCC_ZERO           0x0B20    // Used for ENABLE/DISABLE INTERRUPTS
#define INT_RTCC            0x0B20    // Used for ENABLE/DISABLE INTERRUPTS
#define RB_CHANGE           0x0B08    // Used for ENABLE/DISABLE INTERRUPTS
#define INT_RB              0x0B08    // Used for ENABLE/DISABLE INTERRUPTS
#define EXT_INT             0x0B10    // Used for ENABLE/DISABLE INTERRUPTS
#define INT_EXT             0x0B10    // Used for ENABLE/DISABLE INTERRUPTS

#define GLOBAL              0x0BC0    // Used for ENABLE/DISABLE INTERRUPTS
////////////////////////////////////// Constants used for Timer1 and Timer2
#define T1_DISABLED         0
#define T1_INTERNAL         5
#define T1_EXTERNAL         7
#define T1_EXTERNAL_SYNC    3
#define T1_CLK_OUT          8
#define T1_DIV_BY_1         0
#define T1_DIV_BY_2         0x10
#define T1_DIV_BY_4         0x20
#define T1_DIV_BY_8         0x30
#byte   TIMER_1_LOW=        0x0e
#byte   TIMER_1_HIGH=       0x0f
#define T2_DISABLED         0
#define T2_DIV_BY_1         4
#define T2_DIV_BY_4         5
#define T2_DIV_BY_16        6
#byte   TIMER_2=            0x11

#define INT_TIMER1          0x8C01    // Used for ENABLE/DISABLE INTERRUPTS
#define INT_TIMER2          0x8C02    // Used for ENABLE/DISABLE INTERRUPTS

///////////////////////////////////// Constants used for SETUP_CCP1()
#define CCP_OFF                         0
#define CCP_CAPTURE_FE                  4
#define CCP_CAPTURE_RE                  5
#define CCP_CAPTURE_DIV_4               6
#define CCP_CAPTURE_DIV_16              7
#define CCP_COMPARE_SET_ON_MATCH        8
#define CCP_COMPARE_CLR_ON_MATCH        9
#define CCP_COMPARE_INT                 0xA
#define CCP_COMPARE_RESET_TIMER         0xB
#define CCP_PWM                         0xC
#define CCP_PWM_PLUS_1                  0x1c
```

```
#define CCP_PWM_PLUS_2                  0x2c
#define CCP_PWM_PLUS_3                  0x3c
long CCP_1;
#byte   CCP_1    =                      0x15
#byte   CCP_1_LOW=                      0x15
#byte   CCP_1_HIGH=                     0x16

#define INT_CCP1            0x8C04    // Used for ENABLE/DISABLE INTERRUPTS

//////////////////////////////////// Constants used for SETUP_CCP2()
long CCP_2;
#byte   CCP_2    =                      0x1B
#byte   CCP_2_LOW=                      0x1B
#byte   CCP_2_HIGH=                     0x1C

#define INT_CCP2            0x8D01    // Used for ENABLE/DISABLE INTERRUPTS

//////////////////////////////////// Constants used in SETUP_SSP()
#define SPI_MASTER       0x20
#define SPI_SLAVE        0x24
#define SPI_L_TO_H       0
#define SPI_H_TO_L       0x10
#define SPI_CLK_DIV_4    0
#define SPI_CLK_DIV_16   1
#define SPI_CLK_DIV_64   2
#define SPI_CLK_T2       3
#define SPI_SS_DISABLED  1
#define SPI_SAMPLE_AT_END 0x80  // Only for some parts
#define SPI_XMIT_L_TO_H  0x40   // Only for some parts
#define INT_SSP             0x8C08    // Used for ENABLE/DISABLE INTERRUPTS



#define INT_RDA             0x8C20    // Used for ENABLE/DISABLE INTERRUPTS
#define INT_TBE             0x8C10    // Used for ENABLE/DISABLE INTERRUPTS

///////////////////////////////////// Constants used for SETUP_ADC_PORTS()
#define ALL_ANALOG                0x80
#define ANALOG_RA3_REF            0x81
#define A_ANALOG                  0x82
#define A_ANALOG_RA3_REF          0x83
#define RA0_RA1_RA3_ANALOG        0x84
#define RA0_RA1_ANALOG_RA3_REF 0x85
#define NO_ANALOGS                0x86

#define ANALOG_RA3_RA2_REF              0x88
#define ANALOG_NOT_RE1_RE2              0x89
#define ANALOG_NOT_RE1_RE2_REF_RA3      0x8A
#define ANALOG_NOT_RE1_RE2_REF_RA3_RA2  0x8B
#define A_ANALOG_RA3_RA2_REF            0x8C
#define RA0_RA1_ANALOG_RA3_RA2_REF      0x8D
#define RA0_ANALOG                      0x8E
#define RA0_ANALOG_RA3_RA2_REF          0x8F
///////////////////////////////////// Constants used for SETUP_ADC()
#define ADC_OFF                   0
#define ADC_CLOCK_DIV_2           1
#define ADC_CLOCK_DIV_8       0x41
#define ADC_CLOCK_DIV_32      0x81
#define ADC_CLOCK_INTERNAL    0xc1

#define ADC_DONE            0x8C40    // Used for ENABLE/DISABLE INTERRUPTS
```

```
#define INT_ADC             0x8C40    // Used for ENABLE/DISABLE INTERRUPTS

#list
```

C.4. Host computer calibration routine

Before running the Musical Trinkets program, the frequencies for each tag in the system had to be recorded so the program could distinguish between tags. This program was used for this calibration, and without being entirely too long, gives relevant information on interfacing a Windows PC with the tag reader board.

```
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <iostream.h>
#include <fstream.h>
#include <conio.h>

const TOTALTAGS=20;
int maxes[TOTALTAGS];
int freqstart[TOTALTAGS], freqend[TOTALTAGS];
char tagnames[TOTALTAGS][30]={"red goblin","green goblin","blue goblin",
        "white ring", "red ring","yellow ring","green ring","blue ring",
        "distorting tag","dinosaur","porcupine","pikachu",
        "box1","box2","box3","pig","eye1", "eye2", "eye3", "cone(switch)"};
static HANDLE hCom;
char serialPort[30] = "COM2";
int i, j, k;
int data[256];
int pos=0;
int totalcount=10000;
char filename[80]="c:\\tags.siggraph\\tagcal.txt";
void comSetup();
void calibtag(int *start, int *end, int *max);
void savetofile();
int* read();

void main()
{
        // quick calibration-file-generating hack for the tags system


        // first, read in the current calibration-file
        ifstream infile(filename);
        if (infile == NULL) {
                cout << filename << " could not be loaded!  dropping to defaults...\n";
        }
        else {
                // load in current calibration
                infile >> serialPort;
                cout << "serial port " << serialPort << endl;
                for (i=0; i<TOTALTAGS; i++) {
```

```
            infile >> freqstart[i] >> freqend[i] >> maxes[i];
            cout << "tag " << i << ":" << freqstart[i] << " " << freqend[i] <<
    “ " << maxes[i] << '\n';
        }
    }
    comSetup();
    infile.close();
    cout << "Press key to continue...\n";
    cout.flush();
    _getch();

    int laststart=0, lastend=0, lastmax=0, lasttag=-1;
    int keeplooping=1;
    char input=0, currtag=0;
    while (keeplooping == 1) { // main menu loop
        while (_kbhit()) _getch();
        cout << "Current tag values (name, frequency range, amplitude):\n";
        for (i=0; i<TOTALTAGS; i++) { // print menu
            cout << (char)((i<10)?'0'+i:('a'+(i-10)))
                << ":" << tagnames[i] << " " << freqstart[i] << "-" <<
freqend[i] << " "
                << maxes[i] << endl;
        }
        cout << "Press number/lowercase letter to calibrate tag, or S:save current
calibration,\n"
            << "T:test tag (dump values), U:undo last tag calibrate, Q:quit\n";
        cout.flush();
        input=_getche();
        cout<<endl;
        switch(input) {
            case 'Q':
                keeplooping=0;
                break;
            case 'T':
                while (!_kbhit()) {
                    cout.flush();
                    read();
                    for (i=0; i<pos; i++) {
                        cout << data[i] << " ";
                    }
                    cout << endl;
                }
                break;
            case 'U':
                if (lasttag!=-1) {
                    freqstart[lasttag]=laststart;
                    freqend[lasttag]=lastend;
                    maxes[lasttag]=lastmax;
                    lasttag=-1;
                }
                else cout << "Can't undo.\n";
                break;
            case 'S':
                // calculate optimal frequencies and dump to file
                //  - sort by freqstarts, then average freqstarts.
                //  - if overlap, let lower tag take precedence
                savetofile();
                break;
            default:
                if (input >= '0' && input <= '9' && TOTALTAGS>input-'0') {
                    currtag=input-'0';
                }
                else if (TOTALTAGS > 10 && (TOTALTAGS-10) > input-'a') {
                    currtag=input-'a'+10;
                }
                else {
                    cout << "invalid input.\n";
                    break;
                }
                lasttag=currtag;
                laststart=freqstart[currtag];
                lastend=freqend[currtag];
                lastmax=maxes[currtag];
                calibtag(&freqstart[currtag], &freqend[currtag],
&maxes[currtag]);
                break;
        }
    }
}

void calibtag(int *start, int *end, int *max) {
    int locounts[TOTALTAGS], hicounts[TOTALTAGS], sums[TOTALTAGS], middles[TOTALTAGS];
    *start=10000;
    *end=0;
    *max=0;
    while (!_kbhit()) {
        read();
        if ((pos-4)/4 <= 0) {
            continue;
        }
        for (i=0; i<pos; i+=4)
        {
            locounts[i/4]=data[i];
            hicounts[i/4]=data[i+1];
            sums[i/4]=(data[i+2]<<8) + data[i+3];
            middles[i/4]=(locounts[i/4]+hicounts[i/4])/2;
        }
        totalcount=(data[pos-4]<<8)+data[pos-3];
        if (*start>locounts[0]) *start=locounts[0];
        if (*end<hicounts[0]) *end=hicounts[0];
        if (sums[0]>*max) *max=sums[0];
        cout << (*start) << "-" << (*end) << " " << (*max) << endl;
    }
}

void savetofile() {
    int order[TOTALTAGS], realstart[TOTALTAGS], realend[TOTALTAGS];
    int temp;
    for (i=0; i<TOTALTAGS; i++)
        order[i]=i;
    for (j=0; j<TOTALTAGS; j++) {
        for (i=0; i<TOTALTAGS-1; i++) {
            if (freqstart[order[i]]>freqstart[order[i+1]]) {
                temp=order[i];
                order[i]=order[i+1];
                order[i+1]=temp;
            }
        }
    }
    realstart[order[0]] = 0;
    for (i=0; i<TOTALTAGS-1; i++) {
        if (freqend[order[i]]>=freqstart[order[i+1]]) {
```

```
                        cout << "warning!  overlap between " << tagnames[order[i]] << " and
" <<
                                tagnames[order[i+1]] << endl;
                        realend[order[i]]=freqend[order[i]];
                        realstart[order[i+1]]=freqend[order[i]]+1;
                }
                else {
                        realend[order[i]]=(freqend[order[i]]+freqstart[order[i+1]])/2;
                        realstart[order[i+1]]=realend[order[i]]+1;
                }
        }
        realend[order[TOTALTAGS-1]] = totalcount;

        ofstream outfile(filename);
        outfile << serialPort << endl;
        for (i=0; i<TOTALTAGS; i++) {
                outfile << realstart[i] << ' ' << realend[i] << ' ' << maxes[i] << endl;
        }
        outfile.close();
        cout << "file " << filename << "written.";
        _getche();
}

int* read() {
        DWORD bytesread;
        unsigned char buffer[256];

        pos=0;
        while (1)
        {
//              Sleep(1);
                ReadFile(hCom, buffer, 1, &bytesread, NULL);
//              cout << "read" << bytesread;
                cout.flush();

                if (bytesread)
                {
//                      cout << "read";
//                      cout.flush();
                        for (int i=0; i<bytesread; i++)
                        {
                                data[pos++]=buffer[i];
                                if (pos>250) pos=0;
                                if (data[pos-1]==255 && data[pos-2]==255)
                                {
//                                      cout << endl << pos << endl;
                                        return data; // data in data, pos in pos (globals)
                                }
                        }
                }

        }
}

static
void
comSetup()
{
        DCB dcb;
        DWORD dwError;
        BOOL fSuccess;
        hCom = CreateFile(serialPort,
                GENERIC_READ | GENERIC_WRITE,
                0,    /* comm devices must be opened w/exclusive-access */
                NULL, /* no security attrs */
                OPEN_EXISTING, /* comm devices must use OPEN_EXISTING */
                0,    /* not overlapped I/O */
                NULL  /* hTemplate must be NULL for comm devices */
                );

        cout << "Opened serial port " << serialPort << " " << hCom;
        if (hCom == INVALID_HANDLE_VALUE) {
                dwError = GetLastError();

                /* handle error */
//              r->ew.w("Error opening %s",serialPort);
                return;
        }

        /*
        * Omit the call to SetupComm to use the default queue sizes.
        * Get the current configuration.
        */

        fSuccess = GetCommState(hCom, &dcb);

        if (!fSuccess) {
                /* Handle the error. */
//              r->ew.w("Error getting the comm state");
                return;
        }

        /* Fill in the DCB: baud=19200, 8 data bits, no parity, 1 stop bit. */

        dcb.BaudRate = 19200;
        dcb.ByteSize = 8;
        dcb.Parity = NOPARITY;
        dcb.StopBits = ONESTOPBIT;

        fSuccess = SetCommState(hCom, &dcb);

        if (!fSuccess) {
            /* Handle the error. */
//              r->ew.w("Error setting the comm state");
                return;
        }

        COMMTIMEOUTS cto;
        cto.ReadIntervalTimeout = 4;
    cto.ReadTotalTimeoutMultiplier = 10;
    cto.ReadTotalTimeoutConstant = 100;
    cto.WriteTotalTimeoutMultiplier = 10;
    cto.WriteTotalTimeoutConstant = 100;
        if (!SetCommTimeouts(hCom, &cto)){
//              r->ew.w("Unable to set proper timeouts");
        }
}
```