Hex - a Navigating System for the Blind

Birkir R Gunnarsson

May 5, 2002

Abstract

In this paper I will present the results of my research in designing an affordable navigation system for the blind and visually impaired. I will discuss the original goals of the design and the design problems that had to be dealt with as well as the solutions I came up with for those problems. Finally, I will discuss how this project could be developed further and integrated into a more powerful system which, by exploiting recent advances in imbedded systems and telecommunication technologies, might provide blind people with opportunities for improved spatial perception and more independent inner-city travelling.

Background and Introduction

Recent advances in embedded systems development have opened up a multitude of possibilities in almost every field of information processing. One application of such systems is to facilitate inner city travel for the blind and visually impaired. Such
systems have been referred to as Electronic Travel Aids (ETAs). ETAs have been invented, patented and introduced in many flavors in the last 50 years but none of them has managed to gain widespread use or popularity, often due to high prices, information overload and an unintuitive interface. Also it has been hard for the ETAs to compete with the white cane which, in addition to having approximately 25 different uses is recognized world-wide as being the symbol of the blind. To understand better how an ETA can prove helpful to a blind individual let us now take a look at how most ETAs have traditionally operated and what perception problems they have been designed to solve.

Humans have three spatial sensing mechanisms. Listed in order of range they include binocular vision, binural hearing and active touch. Thus it is the purpose of almost any ETA for the blind to pick up spatial information that other people obtain via vision and translate it into information communicable by ways of hearing or touch. The information traditionally involves some detection and description of objects (laser scanning devices) as well as, more importantly, its location and how far it is from the user (ultrasonic devices such as the Sonic Path Finder) (Milios and Stergiopoulos, 1999). As such the ETA designer faces two problems: Spatial sensing (gathering information from the environment) and interfacing (communicating that information to the user). Past attempts at spatial sensing have included techniques such as ultrasonic air sonar, rangefinding via laser triangulation, and most recently GPS (global positioning system) (Yen, 1996) using radio triangulation of timing signals from simultaneously viewed satellites.

The second and even more challenging problem is communicating the information to the user. The interfacing has to rely on the two remaining sensory systems available i.e. hearing or touch.

To communicate information audially we can use the sound attributes which include the following: 1) loudness (intensity), 2) pitch (frequency), 3) phase (with respect to a reference), 4) timbre (spectrum or equivalently the waveform shape as a function of time), and 5) spatial location (direction and range) Yen (1996). Of these five the first two have traditionally been used for ETA interfacing.

Similarly, our skin can detect vibrational stimuli (changing pressures) with attributes such as: 1) intensity, 2) frequency, 3) phase (when two or more points are stimulated), 4) waveform shape as a function of time, and 5) location (and shape) of stimulation on the skin’s surface. When combined these features allow us to detect the shape an
orientation of an object by passing our skin sensors over the surface of the object. This is referred to as "active touch".

Studies have found that humans rely on their auditory system for inferring distances in their surroundings (echology). Worchein (1947) found that partially deaf and blind subjects could not detect obstacles if audition was prevented. If, however, the skin of the external ears was covered but the auditory meatus was exposed, subjects could still detect obstacles. These findings were further supported by similar studies performed on bats (Griffin and Galambos, 1941).

The biggest challenge in ETA design, as previously mentioned, has been to indicate the distance of the object from the user in an understandable manner. Traditionally changes in pitch and or vibration have been used to provide an idea of the objects relative distance, becoming more intense as the user approaches the object. So far no product that tries to do this has gained widespread use within the blind community. This is probably due to its pricing (generally laser canes and head mounted sonar scans retail between $750 and $2500).

Keeping the above discussion in mind I wanted to try a different approach to ETA design. Seeing that object detection is, at best, difficult and requires expensive hardware and, at worst, inaccurate and confusing, I wanted to think of other ways to aid the blind inner-city traveller by concentrating on information that the white cane cannot provide. There is a wide variety of tasks for such a system as can easily be seen from observing a blind person walking around in a city.

The biggest needs of blind and/or visually impaired people walking around in a city include: being able to navigate through a territory with no obvious landmarks (such as walls, trees, grass lines etc); to locate a particular location (building, bus stop, a certain street); being able to independently use public transportation systems (recognizing a particular bus or train without having to ask a sighted person); and being able to look up information concerning their present location. At this point there still remain technical challenges to designing a reliable and robust system that can perform all these functions but, as this paper will demonstrate, a combination of simple sensors and micro controller systems have the capability to deal with at least some of these issues in an efficient and inexpensive manner.

Due to the time frame and the fact that this has been my first time working with imbedded systems, I chose to concentrate on implementing only the navigation feature mentioned above. The system I designed allows a blind person to walk in a predeter-
mined direction in the absence of notable landmarks. This is a very useful feature when navigating through big open areas and, most importantly, to maintain the right heading while crossing a street. It eliminates the possibility of diverting from the original direction during the crossing and as such helps to prevent dangerous circumstances that can arise if the person diverts too much and starts walking parallel to traffic.

When a user has lined herself up in what she perceives to be the right direction she pushes the button and the system stores the direction. Then as long as the system is turned on it will transmit an audio warning once she is heading in a direction that differs by more than 10 degrees from the initial heading. Then, when she either needs to readjust the direction or has no further need of the system because she has found another landmark to follow the system is simply turned off by another button push. The main advantages to this system is that it is simple and easy to use, does not try to locate and track the position of surrounding objects but only concentrates on the users’ relative heading and it uses inexpensive mass-produced hardware and is affordable.

The main design goals behind the system as outlined in the discussion above were the following:

1. Affordability by using inexpensive hardware.
2. Allowing the user the possibility of resetting the system at will.
3. Reliably detect the compass directions.
4. Having a clearly defined and informative alert level (when current direction differs from the "right" (initial) direction by more than a prespecified term (in degrees).)
5. Alerting the user of both the diversion from cause as well as indicating to her what actions need to be taken (i.e. which way to turn to get back to initial heading).
6. To work reliably in the presence of tilting or at least to alert the user if tilting is enough to make the system unreliable.
7. Portability: to make the system compact, light and convenient to carry.
8. Over-all reliability: extensive outdoor testing of the prototype to ensure the code and hardware work well under real-world conditions.

In the following sections I will discuss the 5 main design issues I dealt with during the implementation process (hardware selection and limitations, interface design, error calculation, user control and testing/constructing a suitable carrying case). Then I will discuss how well my prototype performed using the eight criteria listed above.
In conclusion I will elaborate on how integrating my work into other projects and research could lead to the development of a practical and marketable product which could greatly facilitate inner-city travel for the blind/visually impaired individual and the major challenges faced by those who may undertake such projects.

1 Hardware Selection and Limitations

The core of the system I worked on is the PIC 16f877 Micro Controller from Microchip, traditionally used by Yale’s Electrical Engineering department. Its main advantages are its low price (less than $20) and the fact that it is possible to write its routines using C and then compile it, rather than writing all its code using assembly instructions. However, its limitations include limited storage capacity and extremely limited RAM. Also its low processing power does not allow for using trigonometric functions in real-time calculations which turn out to be a bit of a problem in the interface design.

The other main component of the system is the Vector 2x Compass from PNI. The compass is relatively inexpensive (around $50) but suffers from certain limitations. The most problematic of these turned out to be its inability to operate accurately when tilted. As this project only requires around 10 degrees accuracy it turned out that I could allow for up to 10 degrees of tilting without that affecting the performance of the over-all system (the direction of tilt affects the compasses ability to perform but through experimentation I found that 1.5 degree of tilt roughly corresponded to 1 degree of error). However when faced with greater tilt the compass readings became unreliable and, as such, not useful. Unfortunately I did not have enough time to adequately deal with this problem in the design of the prototype but it can easily be dealt with in the future, either by installing accellerometers or simply by using slightly more expensive compasses that operate more robustly in the presence of tilting. This is important because the instrument will always be subjected to tilting as it would be used by a pedestrian.

The compass communicates information digitally using a 9-bit output and as the following graph depicts can update the micro controller approximately 5 times a second. This delay causes the system to perform poorly in the presence of fast movements which is certainly not a desirable feature.

Some problems were encountered with the digital output of the compass itself which will be discussed in more detail in section 5.
To construct the prototype itself various other non-programmable components such as a speaker and a harness had to be added to make the system easy to carry in an outdoor environment. Over-all the hardware turned out to be good enough to design and test out the desirable functionality of the system but probably not good enough for a real-world usage due to its bulkiness and lack of robustness under real-world conditions.

2 Interface Design

The first coding problem faced in the design of the system was how to communicate information to the user. Given the discussion above I decided to use audio cues rather than vibrations to warn the user if she diverted too much from original direction. The system needs to be non-disruptive and, to avoid interfering with other audio cues, only notify the user when she is off by more than a certain number of degrees. The signal also has to contain some information about how far off the original course she has strayed (i.e. the magnitude of the error). The interface function takes as input an integer value (in the range 0 to 180) representing the absolute value of the difference between the initial direction (in degrees) as captured by the system at the user’s request (see section 4 for discussion) and user’s current heading as updated by the compass.

After experimenting with various pitch forms I decided that the most intuitive way of implementing this feature was the following:
If error as measured by the absolute difference between the set direction and current heading is less than 10 degrees, play no sound. If the error exceeds 10 degrees in either direction the system starts transmitting a signal. The signal’s pitch increases linearly as the absolute error increases up till 60 degrees error has been reached. After that the system plays a high pitched tone indicating that the user has deviated drastically from original direction. Note that the change in pitch rather than the pitch itself conveys the appropriate action to the user. By turning a little bit in either direction the user can deduce from the pitch change how she can locate the right direction again. This also allows the error to be symmetric around the right direction without inconveniencing the user.

To play the pitch I used the Micro Controllers setup timer 2 function which takes three arguments, two constants (set to their maximum value) and an integer value representing the absolute error. The relation between the error (denoted by the iFreq variable) and the frequency of the tone played is:

\[
\text{frequency} = \frac{1}{\text{clock} \times 4 \times 16 \times (iFreq + 1)}.
\]

The system clock is 20 mhz and the maximum value iFreq can take is 255 which corresponds to the lowest pitch the micro controller can generate. Experimentation showed that the highest pitch audible through the speaker we used corresponded to an iFreq value of 135. To give an idea of the frequencies these two numbers correspond to let us now plug these two values into the above formula (approximating mhz by 1,000,000 hz).

\[
\text{lowest frequency} = \frac{1}{20,000,000 \times 4 \times 16 \times (255 + 1)} = 1220.7\, \text{Hz}
\]

\[
\text{highest frequency} = \frac{1}{20,000,000 \times 4 \times 16 \times (135 + 1)} = 2297.8\, \text{Hz}
\]

Figure 3 displays the way the system functions, as described above, using the frequency of the sound played corresponding to the absolute error. Note that the use of the absolute value causes the graph to be perfectly symmetric.

Several alternatives were considered for this interface. One of them involved cycling through all frequencies using a counter for the iFreq variable. On every cycle it would go from 255 down to 0, creating sound oscillation. By varying the period for each frequency range many different types of sound oscillations could be created. The result of these experiments generally sounded like a cross between a Gameboy computer game
and a siren. As amusing as these sounds turned out to be and despite the fact that using these types of sounds much broader range of information could be communicated, I decided against using this method because the sounds were non-intuitive and did not give a clear and immediate indication of the error. These do, however, offer good ways of communicating additional information if the system’s capabilities are expanded in the future.

Another idea was to have the pitch increase as user turns to the right and decrease as she turns to the left. As it turned out, however, the micro controller / speaker is only capable of generating six or seven distinct pitches in the audible frequency and that created a trade-off between communicating the level of the error vs. telling the direction of the error. I decided that it was more important to give the user as detailed information about her direction as possible and thus make the error alert pitch changes the same for both right and left. A better speaker could possibly incorporate enough frequencies to allocate different pitches for different directions.

During the testing process I found that the interface I designed for this system worked and I was satisfied with its implementation.

Figure 3: Frequency Response
3 Error Calculation

To make it easier to invoke the interface function I wrote a function that calculates the corresponding absolute error value. The reason why this procedure has to be written separately, rather than to invoke the interface function simply with \texttt{maketone(current-initial)}; is the cyclic nature of the compass readings (i.e. the fact that we need to account for the borderline cases where compass readings go from 360 to 0 or from 0 to 360. As an example, if the initial heading is 340 degrees and the user keeps turning right the absolute error will reach 20 (360 - 340) but then, instead of being 21 for a one-degree turn to the right, the error will become -339 (1 - 340). We can recognize these borderline cases by the fact that the value of the absolute error is greater than 180 degrees. Furthermore it is easy to account for the error simply by adding or subtracting 360 degrees, depending on which end of the spectrum we are approaching.

Thus we have to deal with three separate cases to make sure that the error is correctly calculated (\texttt{current} refers to current heading and \texttt{initial} refers to the value initialized when the user pressed the "Set Direction" button):

\begin{enumerate}
  \item \texttt{case 1:}
      \begin{itemize}
        \item \texttt{If current - initial < -180 /* described above */}
        \item \texttt{error = error + 360}
      \end{itemize}

  \item \texttt{case 2:}
      \begin{itemize}
        \item \texttt{If current - initial > 180 /* deviating left from initial close to 0 */}
        \item \texttt{error = error - 360}
      \end{itemize}

  \item \texttt{case 3:}
      \begin{itemize}
        \item \texttt{error = abs(current - initial) /* general case */}
      \end{itemize}
\end{enumerate}

Now I could use the value produced by this function as input whenever I wanted to call the maketone function as described in the previous section. This function was extremely simple to implement once I had recognized the borderline cases and dealt with them.
4 Implementing User Control

The last feature that needed to be implemented before the device could be reliably tested was to write a pin-interrupt so that the user could turn the device on and off. This, in fact, serves two purposes.

1. Minimizing the inconvenience to the user of having to listen to redundant sound cues and
2. Allowing the user to set the initial direction whenever she wants to walk straight.

As such this was a fairly simple task and involved writing an interrupt routine that is called whenever the device detects a button push at a prespecified pin. The button push causes a global flag to be set. The value of the flag is being constantly checked by the Main function and as such it determines the actions taken by the system.

When the flag is equal to zero (system is turned off) Hex plays a sound with a very high frequency that is not audible. A more elegant solution would have involved having the system simply wait or remain inactive somehow but it seemed like a good enough solution for a prototype.

When the flag is switched back to one (system turned on) the first thing the code does is to grab the most recent compass reading and set it as the initial variable.

After that it stays in a while loop calculating the error whenever it gets a new reading from the compass and calling the interface function with the absolute value of that error.

The coding concepts were very easy but debugging turned out to be a little bit more difficult because of the PIC’s limited processing power. As it turned out doing anything other than setting the flag in the interrupt function caused inexplicable errors. However, when the actions were moved to another function that problem disappeared. The rest of the code for this feature involved enabling event detection and setting the correct pin. It was easy to write, just referring to the PIC’s manual but very informative as it introduced me to dealing with hardware and coming in contact with the actual device I was programming for. When programming for a typical processor one usually does not need to consider the device’s limitations. Furthermore this introduced me to sottering and hardware debugging.
5 Testing and Integration

Having written the basic functionalities of the system integrating them was a fairly simple task. The code worked as expected. However, the header file for the compass turned out to have significant problems associated with it. Testing revealed that the most significant bit was sometimes not transferred from the compass to the PIC and, hence, the value of the readings never exceeded 255. After consulting the compass’s manual its header file was rewritten from scratch (in collaboration with EE student Alan Ghelberg, who needed to do the same for his project). With the new header file the compass worked and testing showed that in the absence of tilting the readings were constant and fairly reliable. However further testing showed that 1.5 degrees of tilting caused an average of 1 degree error in the readings (as mentioned in section 1).

I experimented with a mechanical circuit to detect and warn the user of excess tilting. The tiltometer consisted of a bell with a pendulum in the middle. When the device was tilted too much the pendulum came in contact with the side of the bell. by passing current through the bell I could determine whether the pendulum touched the side (closed circuit). Upon tilt detection Hex would play an oscillating sound, prompting the user to put the system into a horizontal position. Despite its simplicity this solution did not work due to the fact that the pendulum tended to stick to the side of the bell even when the system was in a horizontal position. Other possible solutions to this problem include using accelerometers to measure the degree of tilting. experimentation with such tilt might lead to the discovery a correction factor, that could be applied to the error calculation to adjust for the tilt. However the fact that the errors vary with the direction of tilt as well as with its angle make it hard to reliably construct such a factor.

Buying more reliable hardware would, of course, be the simplest solution but we should keep in mind that affordability is one of the key factors in the design of a successful ETA. A rule of thumb in the blind community states that ETAs should generally retail for less than 200 dollars. Yen (1996)

At this stage excessive tilting is still a problem with the system prototype. The problem is minimized by the construction of the carrying case which is strapped onto a person's neck and chest, keeping it well balanced (see Figure 4). An additional benefit of such a carrying case is that it is hands-free and thus does not interfere with the user’s sensing and mobility.
At this point vigorous testing of the prototype has yet to be conducted but initial testing shows that the device works reasonably well. The biggest problem with it, beside the aforementioned tilting, is that the compass’s updating frequency is too low causing the interface signals to lag behind, especially when the user is moving rapidly. Another disadvantage of the prototype is its bulkiness and weight, mostly due to the fact that the compass and the PIC separately take up much space and the battery we used is both large and heavy.

Looking at the benchmarks set forth in the introduction, Hex meets the first 5 assessment criteria but does not fully meet the last three. The tilting remains a problem, the prototype is rather bulky and more testing is needed to fine tune the safety constant (currently 10 degrees, as discussed in section 2).

Section - Hex Version 2.0

Rapid development and plummeting prices in the embedded system’s industry may actually make my system much more powerful and affordable in the very near future. Already new versions of both the PIC and compass that I used are on the market and those offer more accuracy and computational power for the same price. An interest-
ing Mechanical Engineering Project would be to integrate the compass and the micro
controller into a single AA battery powered unit. This would be very space efficient
and eliminate the need for a bulky carrying case which is the main disadvantage of the
prototype. Also installing accelerometers to deal with tilting conditions would greatly
improve the current system’s performance. The main advantage, as previously men-
tioned, of Hex is that it does not compete with the white cane but rather seeks to add
functionalities were the cane fails. In order to compete with the cane we need much
more computational power and the ability to recognize data patterns and communicate
information about those patterns to the user.

As an example the current technology may enable us to detect an object near the
individual and even possibly recognize that the object is a car. However, for that
to mean anything to the user the car has to be familiar, e.g. a neighbor’s car would
indicate that the user is on the right path home. Thus we need the ability to scan
the car, recognize the type and the color and realize the fact that the system has seen
the car before to be able to tell the user anything meaningful. Thus, in my opinion,
those who wish to pursue projects in the ETA field should rather concentrate on using
advances in telecommunications and distributed databases to design devices that can
help the blind city traveller find, for example, a bus stop, be able to tell which bus is
stopping at the bus stop or recognize a certain traffic light. Such systems would require
devices to be installed throughout entire infrastructures of vehicles and/or structures
(installing transmitters on busses and traffic signs that could communicate with small
handheld devices such as a PIC with sensors) but there is an increasing demand and
interest in doing so, especially in Europe. In Cheque such systems are already in their
testing phase and other countries and cities in Europe are under increasing pressure
from local blind organizations to take advantage of the technology that is out there to
facilitate city travel. Integrating such functionality into the current implementation of
Hex would not only make it a useful system but practically something that no blind
person in such a city would like to be without.

Thus I encourage anyone who wants to pursue projects in EE or Telecommunications
to consider using my code and building on it to produce something that is not only fun
to design but really makes a difference in the lives of their fellow-citizens.
References


