Abstract— the life of the blind people is very hard, even today in the 21. Century, when there are several things that make their life easier. Their biggest problem is traffic, because people sense the environment mostly through their eyes and sadly, the other organs cannot substitute sight. On the other hand, in traffic one risks the lives of others too. Serious injuries can be obtained by not seeing the obstacles in the homes. Bones can break, or even worse, permanent injuries can occur. In serious cases even the chance of death is present. Our vision is to help in this problem by developing a device and corresponding software. Handling should be easy, dynamic and minimalistic. The solution consists of two parts. The first being the software running on a Windows Phone based mobile phone, and the other one is the custom built DRAC¹ hardware. The hardware contains an ultrasound sensor and two infrared sensors which detect the distance between the objects and the phone, and then the software processes the signal and notifies the user.

I. INTRODUCTION

According to the research made by WHO in the October of 2011, there are nearly 285 million people visually impaired worldwide (TABLE I.): 39 million are blind and 246 have low vision. People 50 years and older are 82% of all blind. The major causes of visual impairment are uncorrected refractive errors (43%) and cataract (33%); the first cause of blindness is cataract (51%). Visual impairment in 2010 is a major global health issue: the preventable causes are as high as 80% of the total global burden.

<table>
<thead>
<tr>
<th>(A) Bl</th>
<th>(B) Low</th>
<th>(A+B) Visually Impaired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population (millions)</td>
<td>Blind (millions)</td>
<td>Vision (millions)</td>
</tr>
<tr>
<td>6,737.50</td>
<td>39.365</td>
<td>246.024</td>
</tr>
</tbody>
</table>

Blindness is the condition of lacking visual perception due to physiological or neurological factors. Various scales have been developed to describe the extent of vision loss and define blindness.[5] Total blindness is the complete lack of form and visual light perception and is clinically recorded as NLP, an abbreviation for “no light perception”. [5] Blindness is frequently used to describe severe visual impairment with residual vision. Those described as having only light perception have no more sight than the ability to tell light from dark and the general direction of a light source.

In many countries the state of legal blindness is defined. In North America and Europe it means ten times worse sight than in the case of healthy subjects. Nearly 10% of the legal blinds cannot see at all.[6]

The life of the blinds – despite the innovations of the XXI. century – is very hard. There were several innovations born to help their everyday lives. For reading and writing, they use Braille writing and languages based on it. Since impaired vision typically occurs in old age, only 10% of the blind knows it. For using computers, there are text to speech and magnifier software. Even audio books are helpful for them.

In traffic, their most important aid is the white cane. This helps them to notice the inequalities on the road in time, and some potential dangers.

The device developed by us is like the white cane, helps them in the traffic. DRAC (Digital Radar Aided Cane) maps the area in front of the user using three sensors and a Windows Phone based mobile phone, which is controlled by voice commands. The phone processes the data from the sensors, and by the frequency of the sound it notifies the user about the distances. The device can detect the threshold. Moreover, the user can request for assistance using SMS gateway service.

Some differences between the white can and DRAC: are presented in Table II:

<table>
<thead>
<tr>
<th>DRAC</th>
<th>White Cane</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-280 cm</td>
<td>typically 120 cm fixed</td>
</tr>
<tr>
<td>71 Hz/cm</td>
<td>N/A</td>
</tr>
<tr>
<td>90 degrees</td>
<td>„1 degree”³</td>
</tr>
<tr>
<td>can not contact solid bodies</td>
<td>may cause damage</td>
</tr>
</tbody>
</table>

II. HANDHELD UNIT DESCRIPTION

The handheld unit or the DRAC Locator, together with the Windows Phone itself, is the core of the project. Its basic function is to provide a distance reading in the direction it’s pointed at and transfer that information to the Phone for processing.

A. Architecture

The DRAC Locator’s main tools are three long range distance measuring sensors mounted on a sensor plate at the front of the device. The sensors are categorized into
two categories. The first two sensors are aimed forward and measure the distance in the direction where the device is pointed. These are called the main sensors. The third sensor is the lone member of its category namely the complementary sensors. This particular sensor is aimed at the floor at an angle of 60 degrees and its role is to detect short obstacles that are easily overlooked and therefore pose a great treat. It’s nicknamed the threshold catcher. The first category comprises of two sensors. The core sensor is the Parallax PING ultrasound sensor. It has great characteristics as its fast, small, energy efficient and has a reliable measuring-range of 280cm which is very impressive. What’s even better that it’s precise enough to measure with cm precision. On the other hand the big drawback of relying solely on this sensor in the previous versions of the project was that it has a very bad refraction rate from surfaces mounted at a sharp angle towards the plane of the sensor (that we call the angle of attack). Practically meaning that enclosing an obstacle at a sharp angle would produce a false impression of free space ahead of the user. To address this issue we’ve added a second main sensor deliberately chosen for its much better tolerance of sharp angles of attack. This sensor is a Sharp GP2Y0A02YK infrared long range distance sensor. Though it has slightly weaker characteristics than the PING it’s still a great sensor and most important of all it provides all important reliability. The third, complementary, sensor is also a Sharp GP2Y0A02YK for it needs to constantly operate with a sharp angle of attack.

The sensors are connected to an embedded development board namely the MikroElektronika StartUSB for pic. We’ve decided to use the StartUSB for pic for numerous reasons:

- It’s widely used
- It comes bundled with all the necessary complementary components greatly reducing the risk of short circuits and other constructional failures
- Thanks to the pre-installed boot loader and the USB communication module modifying the firmware is fast and easy
- It comes with MikroElektronika’s successful MikroC compiler and IDE
- It has both the sufficient execution speed and memory for the task
- the central MCU is a PIC 18F2550 which is fast reliable and has very advanced functions such as multi-level interrupts, sleep mode and a 10-bit ADC module
- And last but definitely not least it’s very cheap

The MCU on the board we’ve used (in the following simply “the board”) has a multitude of pins available both as input and output pins, of which we’ve only had the need to use four. We’ve used two pins to read the analogue output values of the two infrared sensors and then thanks to the built in ADC module we’ve converted them to digital readouts. The third pin was used to both control and read the PING so it was set in effect bi-
directional. Thanks to the PINGs digital output signal no conversion was needed there. Finally the fourth pin was used to transfer the synthesized final data to the phone. All this was put together inside a common flashlight casing which provided a readily made casing with ergonomic shape and a modular build aiding the assembly and also the eventual disassembly of the device for maintenance.

The device is powered from a standard 9V battery attached to a voltage stabilizer to provide a stable 5VDC power.

The device is connected to the phone by an audio cable and an adapter that adjusts the signal levels to the phones audio hardware.

B. Operation

The two main sensors measure the distance from the nearest obstacle in the direction the device is pointed and relay that information to the MCU. The MCU then does the necessary conversion and adjustments and analyzes the two readouts to produce a final readout to be sent to the phone for processing. The readout of the threshold-catcher is treated very much the same way and is also packed into the package sent to the phone.

Measuring with the PING sensor relies on the internal timer of the MCU and the INT0 edge change interrupt. Namely the sensor measures by emitting a burst of ultrasound waves after receiving the activation signal from the MCU, and waiting for the echo to return. The output of the sensor is a simple high signal that begins at the end of the burst and the signal remains high as long as the echo returns, thus meaning that the high-time of the signal is directly proportional to the distance. Knowing the frequency of the internal timer and the speed of sound in air the actual distance could be easily calculated. The only needed addition was the INT0 interrupt triggered by the change of the signal level on the pin controlling the PING.

![Figure 1. The Locator](image)

Measuring with the infrared sensors is quite different. They produce an analogue output. Namely a voltage level that is proportional to the distance measured but the relation is not linear. After realizing that the characteristic output graph of the sensors provided in the datasheet could not be easily described with a formula (6th level
polynomial for satisfying correlation) we’ve decided to use a calibration table we’ve created and produce a simple lookup table based on it. This method proved to be simple, fast and reliable.

The two readouts (main and threshold) are packed into a single package and sent through the audio cable to the phone for processing. Locator can be seen on Fig. 1.

III. THE PHONE SOFTWARE

A. The Core

The class diagram of the DRAC Core project (Figure 2):

![Figure 2. The class diagram of the DRAC Core project](image)

The core is a fairly simple algorithm that was optimized to its minimums to ensure speedy execution since it has to run very often much like an interrupt handler would. The core makes use of the XNA Recorder class that enables it to record input from the microphone port and when the buffer gets full it throws an exception. That exception is handled by the core. The buffer is set to its possible minimum which is 100ms. It allows us to pass data to the rest of the software 10 times a second, which we’ve found satisfactory. The core’s core so to say is a function that analyzes the raw data in the buffer. It works by iterating trough the set of data twice. In the first one it analyzes the data and determines the maximums and minimums of the signal and based on that it determines a middle value that is the basis of separating logical ones from zeros. In the second round it categorizes every reading as either high or low and immediately locates the valid data regions between two terminators. After the second loop the only thing left is to read the bits from the valid regions and take their mean value as the final reading for the given period. Finally the final reading is passed on to the higher levels of the software. The Player class does the opposite. Once a new distance is read, it gets passed the Player objects and it generates a sound based on the input frequency. The formula is easy:

\[
frequency = (MAX\_FREQUENCY \cdot \text{distance} \cdot \text{hertzPerCm})
\]

B. High level module (Placeholder1)

At the time of the design, we wanted to make the software as modular as possible. By this, the high level module – the module which contains the GUI – is rather simple (Figure 3.).

![Figure 3. The DRAC interface](image)
As the Player and Recorder classes do all the work, all the coding needed was to create instances of those objects:

```csharp
Recorder = new Recorder();
Recorder.PropertyChanged += Recorder_PropertyChanged;
Player = new Player();
```

Thanks to the data binding capabilities of WPF, the event handler can be as simple as two lines of code:

```csharp
void Recorder_PropertyChanged(object sender, PropertyChangedEventArgs e)
{
    CurrentDistance = Recorder.DistanceReading.ToString(CultureInfo.InvariantCulture);
    Player.Play(Recorder.DistanceReading);
}
```

If the CurrentDistance string is removed from the GUI, the code becomes even simpler.

**CONCLUSION**

During the development process, we gained lots of experience, and — hopefully — created a device which eases the lives of visually impaired people. From the start, it has been designed for the future user, that’s why the GUI is so simple – sadly the users could not see if it were nicer – and this is the reason for the lack of settings. It does provide an out of the box experience. One just has to plug in the device and without any calibration, registration or configuration he can use it.

**REFERENCES**

[1] Parallax: PING ))TM Ultrasonic Distance Sensor (#28015)