Prototyping a Wireless Attention Acquiring Device (WAAD)

Alfred Jason Kamczyc[1]

Abstract – A large percentage of people are affected with hearing impairments. The most common form to gain the attention of someone not looking is by delivering some sort of audible cue. The goal of this project was to develop and test a small WAAD that delivers both a visual and haptic signal the deaf person can see and feel to help alleviate the aforementioned problem. The initial proof of concept was developed with an ATmega328 microcontroller mounted on an Arduino Uno development platform. Two XBee Pro S2Bs were used for wireless communication between the Arduino and computer base station. LEDs were used to simulate and monitor signal transfer. All code for the microcontroller was written and developed using the Arduino IDE, whereas the X-CTU program was used for Windows to interact with the XBee via USB cable.

Keywords: Arduino, XBee, wireless, deaf, attention acquiring

INTRODUCTION

About 8 million people in the United States can be classified as hard of hearing [1]. This means that normal conversations are difficult to hear even with the use of a hearing aid. This highlights a problem that affects a large percentage of the US population, and poses even larger problems for children with hearing deficiencies. Many parents and guardians of children run into problems when trying to get the attention of the child in order to relay a message. This becomes incalculably more difficult when the child cannot hear audible cues. A quick search online proved that no feasible solutions existed to meet this need. In fact, the only WAAD that had similar capabilities to the device desired was an alert system used in many restaurants to signify table availability. The goal of this project was to create a small wireless attention acquiring device that could be worn by a child. The WAAD would relay some kind of alert through lights and vibration motors. Children could be taught that when this alert occurred, the parent/guardian wanted to relay a message to the child. This solves the problem of a child not paying attention or being at a great enough distance (such as in a park) where visual cues would not be immediately seen.

The WAAD was designed with small children in mind to help the hearing impaired in mostly public situations. It can be frustrating for parents/guardians trying to leave public areas with their child away from their side. It can also help increase the safety of children. For example, kids often run ahead of the group with little to no concern for the world surrounding them. A quick alert can stop a child from running across a street at a busy intersection. Although the WAAD was primarily focused for children, the same concept has many uses outside its original application. For example, firefighters could use it as an emergency evacuation signal. Adventurers could use it to signal each other if ever separated from the group, large industrial plants could use it to signal workers in high noise conditions, and nursing homes could use it as a cost effective notification system for the nurses.

It is important to note that the project completed was simply a proof of concept rather than a retail device. Besides a functioning prototype, the project also provided an in depth report on using XBee transceivers with both the Arduino and Windows platform, a detailed guide on XBee initial set up, and a parts list provided to the Mercer Robotics Club for the simplest way to create wireless communication between the two WAADs.

[1] Mercer University, Electrical Engineering Department, 1400 Coleman Ave, jkamczyc@gmail.com

© American Society for Engineering Education, 2014

The Arduino Uno development board is an open-source prototyping platform. It houses an ATmega328p microcontroller and allows for easy creation of interactive objects [2]. The free to download Integrated Development Environment (IDE) allows for easy programming of the WAAD. The Arduino community is very active through forums and blogs which create a large knowledge base and an example for almost all applications. The platform was chosen because of pre-existing familiarity with the C/C++ languages and the extensive examples available on the internet.

MATERIALS AND CONFIGURATION

Two XBee Pro S2Bs were the foundation for wireless communication. The XBee Pro is a wireless transceiver that runs on the ZigBee communication protocol [3]. The ZigBee communication protocol is a standard used for these small transceivers much like wireless routers use the 802.11 standards. Each XBee must be connected to a PC through a USB adapter and initialized through the proprietary X-CTU software. Initial set up configures the basic settings required for the two XBee transceivers to communicate. This configuration includes, but is not limited to; network ID, power settings, baud rate, etc. Figure 1 below shows an XBee mounted on a USB adapter.

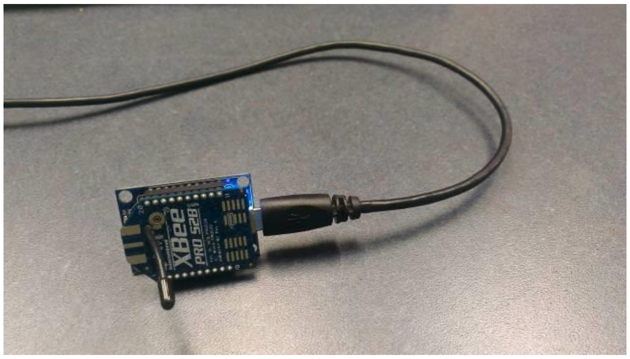

Figure 1. XBee mounted on USB adapter for communication with PC.

After initial set up, one XBee was placed on an Adafruit breakout board which allows for XBee placement on a breadboard and therefore easier connection to an Arduino development platform. The second XBee stays connected to the PC through the USB adapter. Figure 2 shows the final set up of the standalone WAAD fully connected.

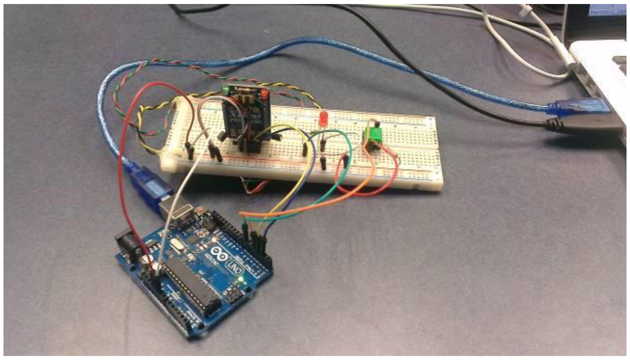

Figure 2. Standalone WAAD connected to PC for programming.

With both XBees initialized, code was developed for the standalone XBee using the Arduino IDE. The program was loaded onto the ATmega328 microcontroller through a USB cord, and the Arduino was powered off a 9 volt battery to demonstrate the mobile capabilities. The XBee connected to the PC was monitored and manipulated using the same X-CTU software that was used for configuration. The computer was used to send messages through the connected XBee in the form of numbers. The Arduino was programmed to receive 4 unique messages denoted through the numbers 0 through 3. The 0 code was the stop message that ended all LED activity, the 1 code produced a slow blink on the LED, the 2 code produced a medium paced blink on the LED, and 3 code produced a rapid blink

© American Society for Engineering Education, 2014

on the LED. The standalone WAAD was also able to communicate back to the PC base station through a button wired to the Arduino. The button can be seen in Figure 2 on the right side of the breadboard. With a simple button press, the standalone WAAD sent a simple message back to the base station that was monitored through the X-CTU software.

TESTING AND RESULTS

Simple wireless communication was tested and provided excellent results. Each code was received by the standalone WAAD with little to no noticeable lag. Figure 3 below shows the LED reacting to the slow blink code.

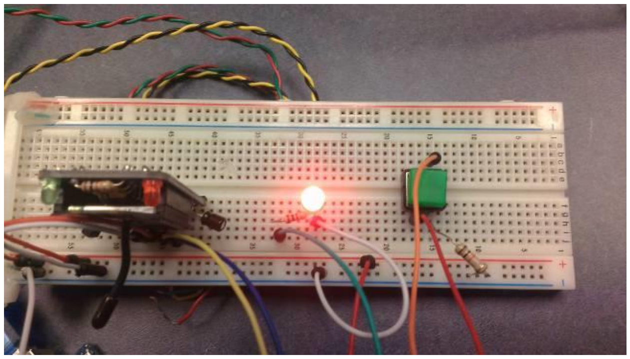

Figure 3: Standalone WAAD successfully receiving commands wirelessly

Two way communication tests were also successful with the base station receiving wireless messages from the standalone XBee with no noticeable lag. Figure 4 below is a screenshot of the X-CTU software successfully receiving the preprogrammed message (red text) and successfully sending out different blink commands (blue text).

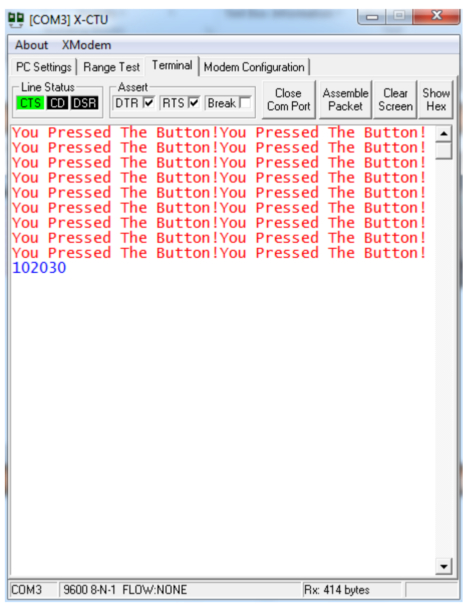

Figure 4: Successful two-way communication shown in the X-CTU program

Wireless strength tests were also conducted to test the range at which the XBee Pro S2Bs could successfully communicate. The first test was conducted in an open space, intramural field, to test the maximum distance at which the two transceivers could communicate. While the specifications claim a distance of 100 meters for wireless communication, the transceivers stopped working reliably around 80 meters. At this distance blink codes started to drop and messages were not received at a 100% success rate. While the test showed the transceivers not reaching

© American Society for Engineering Education, 2014

their advertised distance, 80 meters should be an acceptable working range especially for child tracking applications. The transceivers were also tested indoors in the Science and Engineering Building at Mercer University, a metal frame building. The base station was positioned in a student work lab in the back left corner of the building and was monitored as the standalone WAAD was moved throughout the building. Figure 5 below shows a floor plan of SEB and where the WAAD was tested.

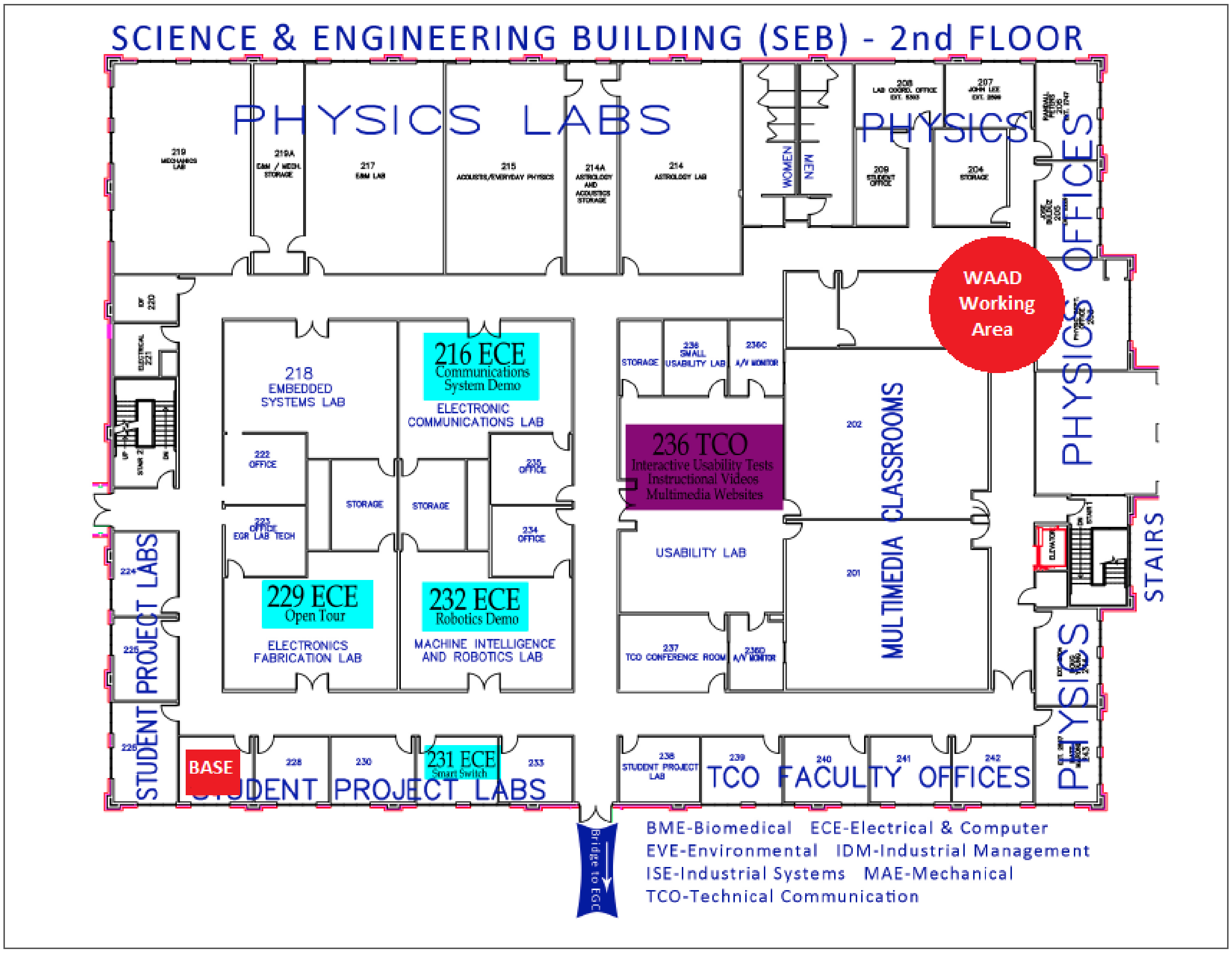

Figure 5: SEB floor plan highlighted with wireless working area.

The interference of the building significantly reduced the working range of the transceivers. Reliable connections started to drop when the standalone unit was transported to the opposite side of the building (a distance of approximately 50 meters). While some messages and codes were received between the two WAADs at this distance, a reliable connection could not be made. This again should not be a concern as the working distance should be acceptable for child tracking in most indoor applications. During testing of the prototype, it should be noted that at no time did any signal interference occur. The WAAD was never activated by interfering wireless signals or other XBee units being used at the same time.

FUTURE WORK

Size Reduction

While the proof of concept is a great starting point, it is far from a completed, marketable device. As seen in Figure 2, the Arduino and breadboard standalone WAAD cannot easily be worn by a small child. In order to make a reduction in size, an Arduino Mini can be used in place of the Arduino Uno. Figure 6 shows an Arduino Mini which

© American Society for Engineering Education, 2014

is about the size of a small USB flash drive. The Arduino is also not the only platform that should be considered. Other options such as the Adafruit Trinket or the Adafruit Gemma are other examples that could be used. While they are not perfectly compatible with the current Arduino IDE, these options provide an even smaller footprint than the Arduino Mini. The Gemma is also designed to be worn on clothing or sewn into fabric. These platforms are all significantly smaller than the Arduino Uno which is about the size of a deck of playing cards.

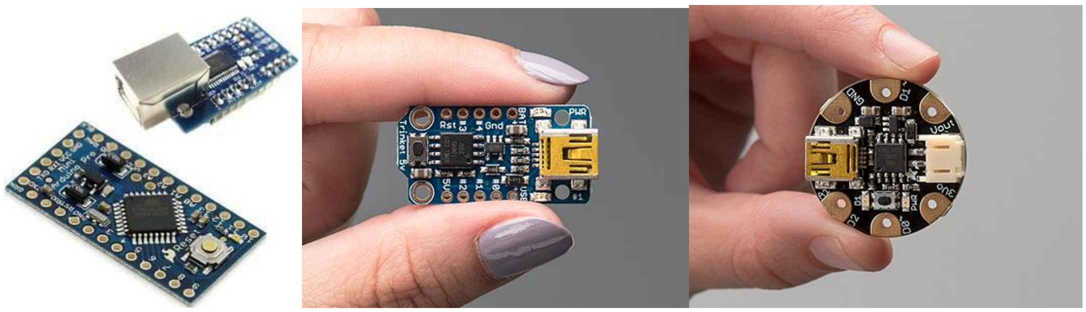

Figure 6: Arduino Mini[4], Adafruit Trinket[5], and Adafruit Gemma[6].

Also, custom printed circuit boards can also be created with relative ease using the available resources at Mercer University. These custom boards will replace the bulky breadboard and house the feedback components. A lithium ion battery should also be used in place of the low power density and bulky 9 volt battery used for demonstration purposes.

WAAD Communication

The demonstration of this WAAD required the use of a computer base station. The base station is quite helpful with respect to code development and debugging tools, but infeasible when considering portability. The Wireless Attention Acquiring Device (WAAD) needs to be developed to work between two microcontrollers instead of one microcontroller and one computer. In order to recreate the 4 unique messages delivered by the base station, more tactile switch could be added to signal different messages. For example four different button presses could represent the four unique states described previously.

Power Consumption

The design currently has both transceivers constantly scanning for signals which accounts for the great response time on both units. This however leads to power consumption issues as the transceivers and Arduino platform constantly need power in order to perform at this level. Each XBee has sleep mode that can be activated to effectively reduce the power consumption when idle for an extended period of time. This feature was turned off as it caused difficulties in simple, close wireless communication at the start of this project. The sleep mode of the XBee needs to be explored more in depth as the Arduino can also be put in a sleep mode and be awoken through the XBee signal. With this mode activated, much of the power consumption issues are resolved. As mentioned before, a lithium ion battery would also be incredibly helpful in regards to replacing the 9 volt battery. It would allow the WAAD to store more power while fitting into a smaller form factor.

Added Features

Haptic feedback (tactile alerts) is a critical feature for signaling the hearing impaired. Even with ultra-bright LEDs, visual warnings can be over looked. A small vibration motor, such as the ones in cell phones, could easily be implemented as a second way to signal the WAAD.

Environmentally hardening the WAAD against water and shocks in particular would result in a much more marketable device. This would also improve the potential for other groups, such as firemen and the outdoor adventurers, to adopt use of the WAAD.

The WAAD could also benefit from a received signal strength indication. While this cannot predict the exact distance the WAADs are from each other, it can warn the parent/guardian of a weak signal which can cause message

© American Society for Engineering Education, 2014

loss. The XBee has a built in pin that indicates received signal strength, but only in the form of good packet or bad packet. A light on the Adafruit breakout board will light up indicating successful transfer or stay dim indicating an unsuccessful transfer. The X-CTU software has the ability to provide the dB strength of the received signal, but provided inconsistent results during testing. Further investigation could lead to accurate signal strength readings greatly increasing the functionality of the WAAD.

Finally, a mesh network of these WAADs could also help in a family situation. Mesh networking allows for the communication of multiple ZigBee based transceivers. This in effect could also increase the range of the WAAD. Mesh networking allows the messages to be relayed through multiple WAADs to reach further endpoints. For example if the child was 120 meters away from the source signal, the person wearing the WAAD would not be warned. However if there was a third WAAD 60 meters away from either WAAD, the message could be relayed through the third WAAD to the original destination effectively increasing the range 40 meters.

Discussion

This project demonstrated the feasibility of using an ATmega328 microcontroller and ZigBee network communication to fabricate a wireless attention acquiring device. Future work on this project should include addressing power consumption issues, implementing a loss of signal notification, adding haptic feedback, and environmentally hardening the device. Once these tasks have been completed, a user testing protocol should be developed.

The XBee devices are considered relatively easy to set up, but the resources available for this device is often very task specific and largely useless if not using the exact same set up as the examples.

References

[1] Mitchell, Ross E., *Unemployment, higher education, and deaf & hard of hearing people: A quick snapshot of research findings*, Gallaudet Research Institute, 2005

[2] Arduino Uno R3 Specification Sheet, https://www.sparkfun.com/products/11021

[3] XBee Pro S2B Specification Sheet, https://www.sparkfun.com/products/10421

[4] http://image.made-in-china.com/43f34j00YZitznRIODqV/Arduino5V-16-MHz-PRO-Mini-Kits-with-Atmega328-Arduino-Compatible-Mini-Kits-Arduino-Uno-Atmega328-MacBook-PRO.jpg

[5] http://www.adafruit.com/images/medium/1501_MED.jpg

[6] http://www.adafruit.com/images/medium/1222_MED.jpg

Acknowledgements

This work would not have been possible without the financial and administrative support of the Engineering Honors Program at Mercer University. I would like to thank Dr. Philip McCreanor, Director of the Engineering Honors Program, for his guidance in developing, implementing, and documenting this project. I would also like to thank Eric Daine and Derek Munday for their help in developing and testing this platform.

Alfred Jason Kamczyc

Alfred Jason Kamczyc is a dual enrolled undergraduate/graduate student attending Mercer University. He is currently pursuing his bachelor's and master's degree in electrical engineering expecting to graduate in May of 2014. He has been involved with many research projects including an external heart rate monitor and autonomous lawn mower. His previous research project in laundry detergent properties received third place for first and second year engineering at the ASEE-SE Section Annual Conference Student Poster Competition held at the Citadel in 2011. His academic success has been noted through Dean's list recognition and many academic scholarships.

© American Society for Engineering Education, 2014

APPENDIX A: ARDUINO CODE

```
#include <SoftwareSerial.h>
// For the electronic wiring , you should :
// Connect pinRx to the Pin2 of XBee(Tx , Dout)
// Connect pinTx to the Pin3 of XBee(Rx , Din)

// Define the pins on Arduino for XBee comminication
long BaudRate = 9600 ;
char GotChar;
int LED = 7;
int button = 4;
int pressed = 0;
unsigned long rssiDur;
int rssiPin = 6;
int samples = 0 ;
// Initialize NewSoftSerial
SoftwareSerial mySerial( 2 , 3 );

void setup()
{ // You shall see these messages in Arduino Serial Monitor
// This part is the official library , it will be used for talking to
// PC serial port
  Serial.begin(BaudRate);
  Serial.println("XBee Communication Test Start!");
  Serial.print("BaudRate:");
  Serial.println(BaudRate);
  Serial.print("NewSoftSerial Rx Pin#");
  Serial.println(2,DEC);
  Serial.print("NewSoftSerial Tx Pin#");
  Serial.println(3,DEC);

  pinMode (rssiDur,INPUT);
  pinMode (LED,OUTPUT);
  pinMode (button,INPUT);

  // This part is the NewSoftSerial for talking to XBee
  mySerial.begin(BaudRate);
  mySerial.println("Powered by NewSoftSerial!");
}

void loop()
{
 pressed = digitalRead (button);

 if (pressed == 1) {
   mySerial.print("You Pressed The Button!");
   }

// Monitor Rx from PC , if the data is available then read
// it to "GotChar".    Then ask XBee send the data out
// wirelessly.
  if ( Serial.available() ) {
      GotChar = Serial.read();
      Serial.print(GotChar);
```

© American Society for Engineering Education, 2014

```
      mySerial.print(GotChar);
  }
// Monitor data from XBee , if the data is available then
// read it to "GotChar".    Then send it back to PC.
  if ( mySerial.available() ) {
     GotChar = mySerial.read();
    Serial.print(GotChar);
      //Serial.print(GotChar);
  }

  if ( GotChar == '1') {
    digitalWrite (LED,HIGH);
  }

 else if (GotChar == '2') {
    digitalWrite (LED,HIGH);
    delay(500);
    digitalWrite(LED,LOW);
    delay(500);
  }
 else if (GotChar == '3') {
    digitalWrite (LED,HIGH);
    delay(100);
    digitalWrite(LED,LOW);
    delay(100);
  }
      else {
    digitalWrite (LED,LOW);
      }
}
```

© American Society for Engineering Education, 2014