The Tracker:
A Proximity Alarm System

Derek Van Riper

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

December 2014

Table of Contents

List of Tables and Figures

Tables:

Figures:

Abstract:

This report covers a proof of concept for a portable device consisting of a microcontroller and Bluetooth module. This device connects to smartphones via Bluetooth signal. Whenever the smartphone loses connection with the device, an alarm via smartphone application alerts the user that they have lost track of the device. The device loses connection with the smartphone when the device and smartphone are separated by more than 30 yards. The device has different proximity settings to allow the user to adjust the critical proximity distance.

Chapter 1: Introduction

Smartphones are essential for young people like myself. In the ages 18-34 demographic, penetration is nearly 80% and the main limiting factor is mostly money. The goal of this senior project is to tap into a lucrative smartphone market by introducing this new product called The Tracker. It will be fun to use, easy to implement, and it will keep track of whatever you attach it to.

Maybe you are the person who constantly forgets to grab their bag before they leave the gym. Or maybe you always forget to grab your leftovers from your friend's fridge before you take off for the night. The Tracker can solve problems like these, and can even be applied to so endless situations. For example, when working on a public computer, it is necessary to use a flash drive to backup your work. However, if a student or professional forgets to take his or her flash drive out of the computer at the end of his or her session, it could be a huge problem for the company or student involved since private information can get leaked in this way.

This device is not meant for users to be able to locate their belongings. Rather, it is meant to simply help individuals remain mindful of where they set down their important possessions, and to remind them when they walk too far away from said possessions. This will hopefully eliminate some of the forgetfulness of individuals like myself, who constantly lose track of important things.

This device will come with a Smartphone mobile application that will alert the user when they lose track of their possession via Smartphone notification. Similar devices have already been designed, but none are tied to a mobile application.

Chapter 2: Specifications and Requirements

Customer Needs Assessment:

Potential customers for The Tracker include all people who regularly use a smartphone. Therefore, the main market for this device targets the majority of the population today. The Tracker will be readily available and anyone can easily sync the device with his or her phone. With only a turn on/off switch on the device, and a sync button, using The Tracker will be a truly intuitive experience. Along with the general population the device would also be very useful to public institutions such as Universities and libraries. Universities could implement The Tracker into their classes to assure the students do not lose important flash drives and projects. The Tracker could also be utilized in classes so that instructors can be notified when a student has left his lecture. Libraries could also use The Tracker to track expensive equipment lent out. If the technology could be advanced more, and the price point dropped, libraries could put a tracker on every book so that they know when books leave the library. The Tracker will make people's lives easier, save them a lot of stress, and will also be fun to use.

TABLE I

THE TRACKER DEVICE REQUIREMENTS AND SPECIFICATIONS

Marketing Requirements	Engineering Specifications	Justification
1,2	Alerts the user that The Tracker has been out of range for 30 seconds via smartphone app	The user will be subject to losing his or her item if The Tracker has gone out of range of the smartphone, but users should be allowed to leave their items unattended for only 30 seconds
5	Battery will last for at least 8 hours and will charge via wall outlet	Long lasting battery will allow the device to be used all day and to be recharged at night
3	Device will be able to communicate with smart phone within a distance of 30 yards	Will allow user to distance himself from The Tracker a small ways before the smartphone app begins to alert
4	Device should be no larger than 2.5” x 2.5” x 1” with 1 on/off switch and 1 sync button	Easy to hide in bags and clip onto belongings
6	Device enclosure should be strong enough to survive a 15ft drop test	If device were to be dropped, it should not break
Marketing Requirements 1. The device must be easy to use and operate 2. The device must have Bluetooth features/capabilities 3. The device must be able to communicate with smartphones 4. The device must be small 5. The device must have a long battery life 6. The device must be durable 7. The device must be intuitive for the user		

Requirements and Specifications:

Table I reveals the full list of market requirements and specifications. The requirements and specifications for this project revolve around the device being reliable, easy to use, and compact. Since this device is being used as a safety precaution, reliability of the product is the most important requirement. This follows the marketing requirements of 3, 5, and 6 that ensure the device is able to durable as well as last for at least 8 hours of use. The second most important requirement is to make it a very small design that someone could comfortably place in his or her pocket without or duffle bag. This is a device that will be carried around with the user, so making it as compact and as non-bulky as possible is a key component to making this device marketable. This follows marketing requirement 4. Lastly, a friendly user interface is important in making this device usable to the everyday person. The goal is to make a device with a single on/off switch that will eliminate any fear of operating from a non-technological user. The user will be able to easily sync the device with their smartphone, turn the device on and it will be ready to use. Requirements 1 and 2 assure that the device's Bluetooth functionality is easy to use and operate.

Chapter 3: Functional Decomposition

Figure 1 shows the level 0 block diagram of The Tracker device. Since one of the main marketing requirements of this device is to be user friendly, the less buttons present the better. An on/off switch will be present for the device to be turned on and off. A second button, the sync button, will be used to sync up the device with the supervisors smartphone. Once the device is synced up and turned on, it will be ready to use.

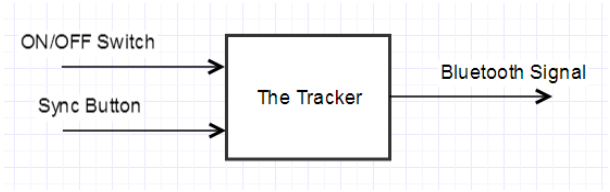

Figure 1: Level 0 Block Diagram

Figure 2 shows the level 1 block diagram of The Tracker. The main idea for this project is to use a Bluetooth peripheral connected to a microcontroller to connect to peoples' smartphones. Once the smartphone loses connection with The Tracker, a timer will start to count. If the timer reaches a predetermined value, indicating that the user has been away from The Tracker device for an amount of time, the smartphone will generate an alert to the user indicating how long ago the item was lost. With this information, it is easy to keep track of anything.

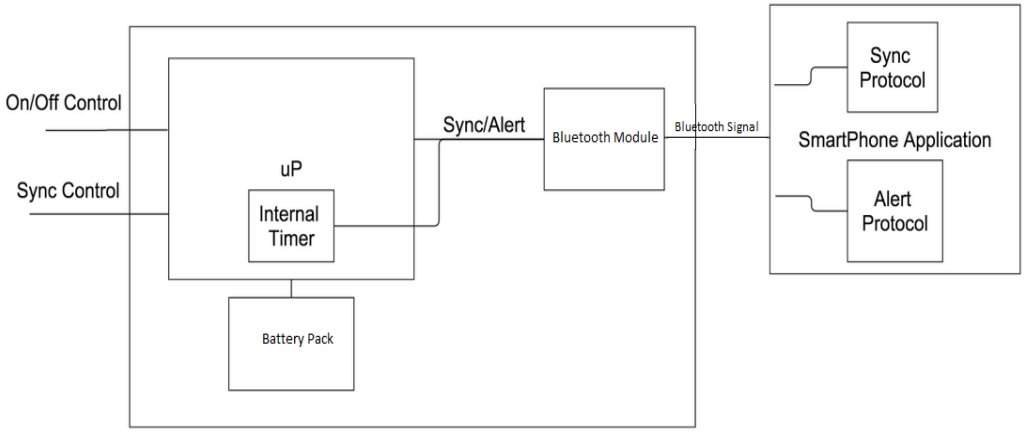

Figure 2: Level 1 Block Diagram

Chapter 4: Design Iteration for Proof of Concept

The design of this project will be powered through USB via computer, and therefore will not be portable. This design iteration will simply act as a proof of concept. From here, I will be able to expand on the design by choosing smaller, more expensive components and implementing a power supply such as a battery pack. The components used in this design step were an Arduino Uno board, a Bluetooth Mate wireless Bluetooth module, and an Android smartphone. The individual components are discussed in detail in the following appropriate sections of this chapter.

Arduino Uno R2

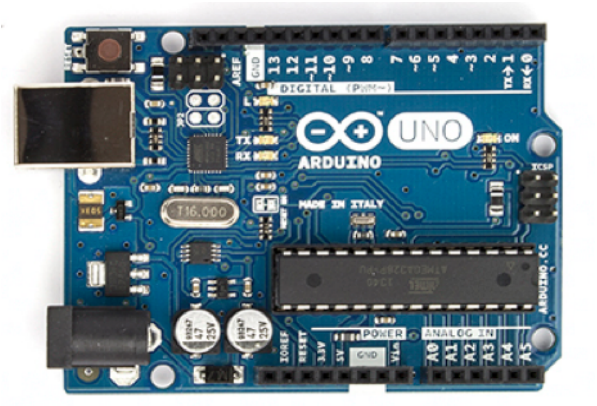

Figure 3: Arduino Uno R2 Microprocessor

Figure 3 shows the Arduino Uno as a microcontroller board based on the ATmega328. The design involves knowledge of embedded system design and analysis. I will be using 2 of the 14 available I/O pins on the board, and the 5V and ground pins. The purpose of using the microcontroller is so that it can initialize and give commands to the Bluetooth module. I am using the Arduino Uno R2 for this design iteration because I already have the board from a previous class. A much smaller microcontroller would be used in the device when the design is to be finalized. Smaller microcontrollers are available to use, but are more difficult to program due to interfacing issues with computers. See Chapter 7 for more details.

Bluetooth Mate Silver RN-42

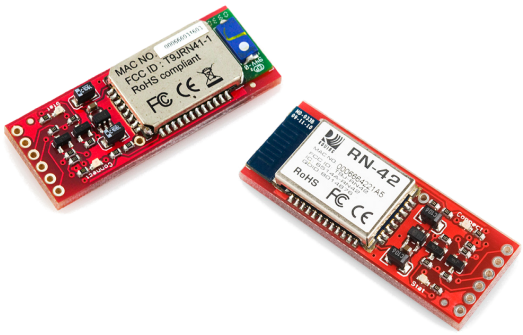

Figure 4: Bluetooth Mate Silver RN-42

Figure 4 depicts the RN-42, which is perfect for short range, battery-powered applications making it perfect for this design. It uses only 26uA in sleep mode while still being discoverable and connectable. Multiple user configurable power modes allow the user to dial in the lowest power profile for a given application. However for this design, I will be focusing on complete functionality at the full power setting. Applying power settings in future design iterations via a smartphone application could allow the user to change the power settings on their tracker directly from their smartphone. This would require additional coding on the microcontroller. By changing the power settings of the Bluetooth module, the user can essentially change the distance from his or her Tracker at which they would like to be notified. This distance is known as the critical distance.

The pinout of the Bluetooth mate is a standardized pinout for a serial interface and power supply combination. This pinout allows the Mate to be plugged directly into the header of most microcontrollers such as the Arduino Pro's and Pro Minis.

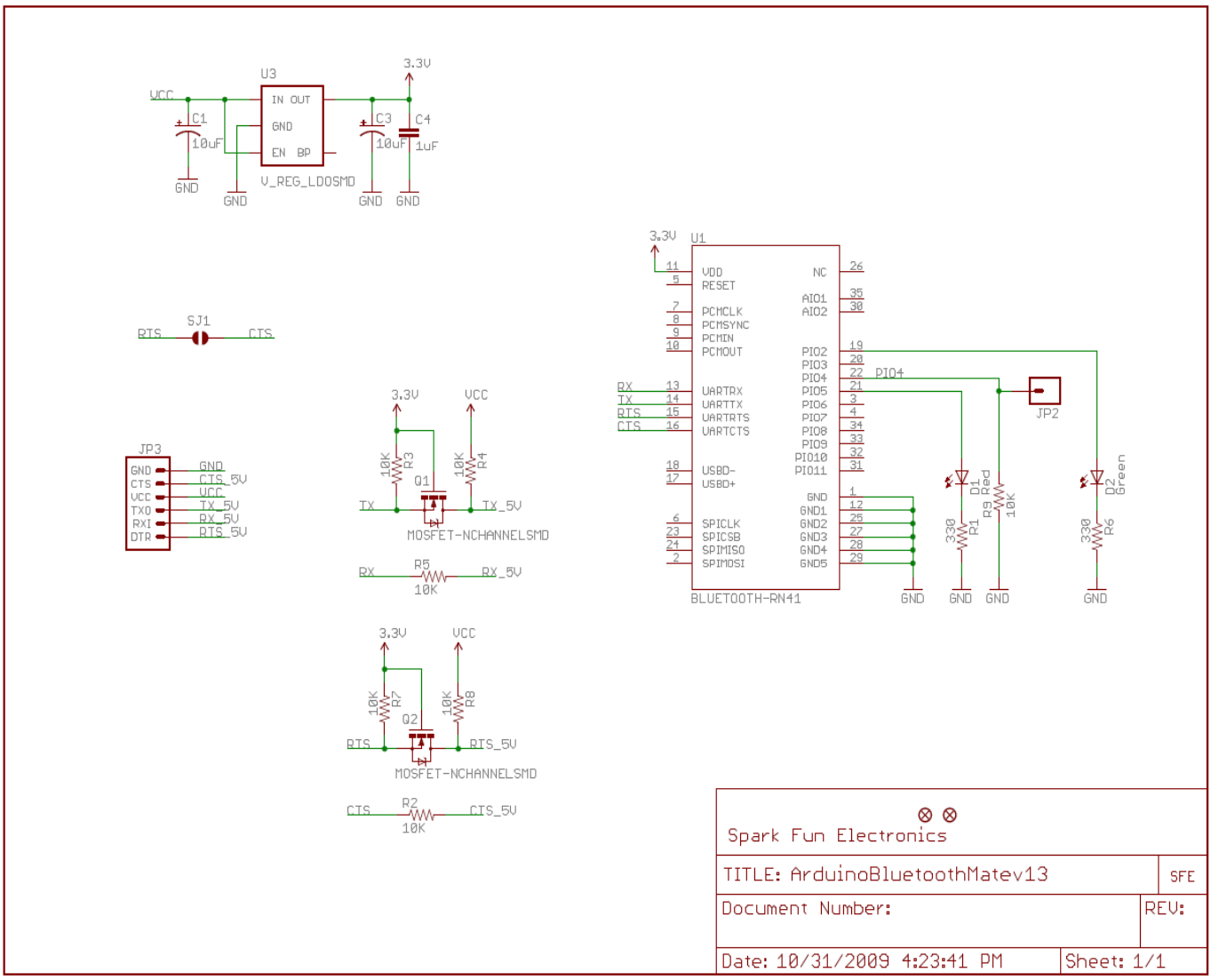

Figure 5: Bluetooth Mate Silver RN-42 Electrical Schematic

Figure 5 shows the electrical schematic reviewed when designing The Tracker. A critical part of the Bluetooth mate is the level shifting circuits between the serial pins, and the output header. The maximum operating voltage of the Roving Networks modules included in the Bluetooth Mate Silver RN-42 is 3.3V. However, the level shifting circuits allow us to use the 5V output from the Arduino Uno to safely communicate with the Bluetooth mate.

To see a diagram of the level shifting circuits, see the voltage regulator circuit depicted in the upper left hand corner of Figure 5. This regulator takes in a higher voltage and converts it to 3.3V, so the device does not get damaged. The regulator can take up to 6V.

TABLE II

BLUETOOTH MATE SILVER RN-42 PINOUT/DESCRIPTION

Pin Label	Pin Function	Input/Output	Description
RTS-0	Request to send	Output	RTS is used for hardware flow control.
RX-1	Serial receive	Input	This pin receives serial data from the microcontroller.
TX-0	Serial transmit	Output	This pin sends serial data back to the microcontroller.
VCC	Voltage supply	Power in	This voltage supply signal is routed through a 3.3V regulator, then routed to the Bluetooth module. It should range from 3.3V to 6V.
CTS-I	Clear to send	Input	CTS is a serial flow control signal.
GND	Ground	Power in	The 0V reference voltage, common to any other device connected to the Bluetooth mate.

Table II depicts each pin and how it is utilized within the Bluetooth Module. This design uses only the TX-0 pin, RX-1 pin, VCC pin, and GND pin. The CTS-I and RTS-0 pins are used for hardware flow control, which is not needed for this design.

TABLE III

BLUETOOTH MATE SILVER ELECTRICAL CHARACTERISTICS

Parameter	Min	Typ.	Max.	Unit
Supply Voltage (DC)	3.0	3.3	3.6	V
Average power consumption				
Radio ON* (Discovery or Inquiry time)		40		mA
Connected Idle (No Sniff)		25		
Connected Idle (Sniff 100 milli secs)		12		
Connected with data transfer	40	45	50	mA
Deep Sleep Idle mode		26		uA

From Table III, it can be derived that the current can be as low as 0.026mA when the device is asleep, and as high as 50mA when the device is transmitting data. These current levels are good for battery powered applications, and should work for portable future designs.

The Bluetooth mate has 2 modes of operation. They are known as command mode, and data mode. To enter the command mode from any data mode, the host controller needs to send a string of three $ symbols. While in command mode, the Bluetooth module can configure characteristics such as device name, baud rate, PIN code, and data rate. Also, action commands such as connect to a device or scan for other modules can be processed in this mode. In data mode, any data received over the Bluetooth connection is routed out the TX pin. Furthermore, the module's RX pin outputs to the device it is connected to via the Bluetooth connection.

The module also has a configuration timer, which is a major obstacle to consider. When the Bluetooth module is turned on, this configuration timer begins. Once the timer is done counting, you'll be unable to enter configuration mode unless you cycle power. By default the configuration timer is set to 60 seconds, however this can be adjusted or even turned off.

Hardware Connectivity

Figure 6 reveals the connectivity of components. The Arduino Uno R2 will connect to the Bluetooth Mate Silver RN-42. Head wires need to be soldered to the Bluetooth module in order for the connection to be reliable.

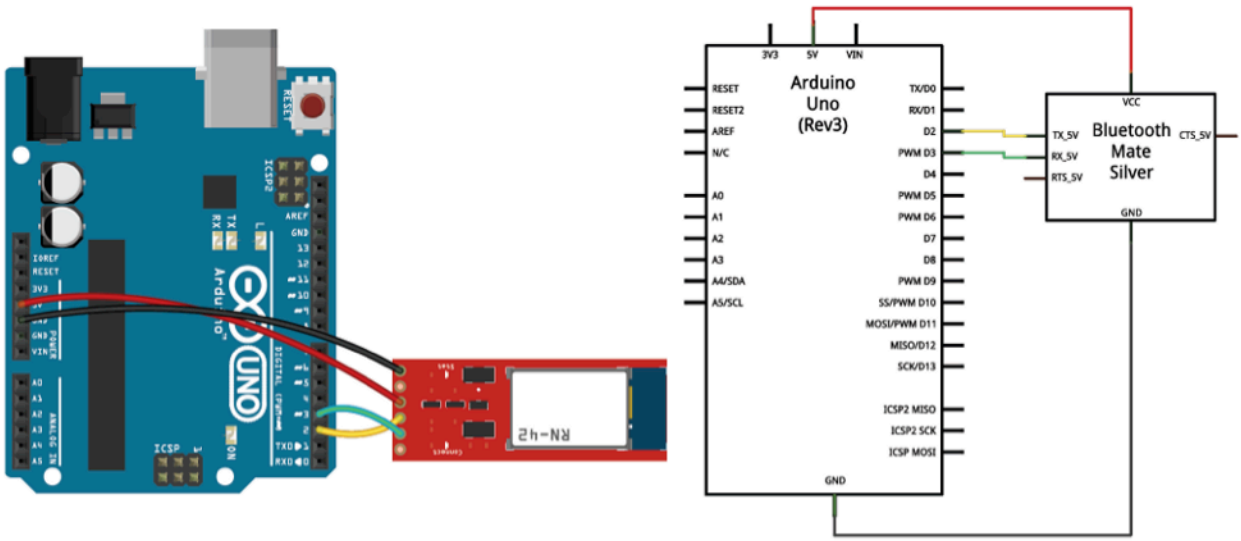

Figure 6: Connectivity of components

Chapter 5: Proof of Concept Implementation and Testing

After connecting the components together and plugging in the Arduino Uno to a computer, you can use the Arduino as a medium between the user and the Bluetooth Mate to send and receive commands. Appendix B at the end of this report contains a section of example code that I used to relay data between the Arduino Serial Monitor for Windows and the Bluetooth modem. Using this code, I was able to connect to the Arduino Uno with my computer, and could communicate with the Bluetooth Mate.

At the beginning of the code, the Arduino enters command mode by entering "$$$" and temporarily changes the Bluetooth modem's baud rate to 9600 bps. This is temporary so that when power is cycled, the Bluetooth mate will return back to a baud rate of 115200 bps.

The loop of the sketch checks to see if the Bluetooth module or the Serial Monitor sent any data to the Arduino. If so, it will relay the data sent from one device to the other.

With the code uploaded to the microcontroller and everything properly connected, I opened up the serial monitor. I first entered command mode by typing "$$$" and clicked Send. This forced the Bluetooth monitor to respond by returning "CMD", indicating that the device is in command mode. From here, I want to scan for discoverable devices, by using the inquiry scan command. To do this, type "I" and click Send. This will return the information as "BT address, BT name, COD", where COD is the class of device.

When the Bluetooth modem discovered my smartphone, I connected it to the Bluetooth mate by using the connect command. To do this, I typed "C,<address>", where <address> is the BT address returned by the inquiry command above.

The device will respond with "TRYING", which will be followed by either "CONNECT failed", or the connection will be successful. After a successful connection the device immediately enters data mode, and the modem becomes a pipeline. This means, any characters sent from one Bluetooth device will be sent to the other, and vice-versa.

After I connected the device to my smartphone, I tested the limits of The Tracker by distancing myself from the device while I was holding my smartphone in hand. I noticed as I got about 30 yards away from the device, the smartphone would lose connection with The Tracker. This distance of 30 yards was without any walls between the device and myself. So with walls and other obstacles, the device may lose connection before 30 yards. Also, if the Bluetooth module were to be set to a lower power level, the distance at which the device may lose connection will also decrease. This leads to various appealing customization settings for users. Ideally, the smartphone application will have the ability to change the Bluetooth mate between various power settings, depending on the user's intended use of The Tracker.

The next step is to create or find a smartphone application that will alert the user via vibration or notification messages in the OS that he or she has lost track of an item. The application will also allow the user to name his tracker so that the application can better identify what item has been lost. For example, if the user is out of range of his car keys, the application would cause the smartphone to vibrate and a notification from The Tracker smartphone application would notify the user "You have lost track of your keys".

Chapter 6: Pairing of Smartphone Application

The smartphone application is necessary to alert the user that he or she has lost contact with The Tracker. The application needs to be very simple, intuitive, and lightweight. For the purposes of simply proving the concept, I opted to focus on an Android application, since I own an Android smartphone. Developers of Android applications use an open source "BluetoothAdapter" class to communicate to and from the local Bluetooth module within the smartphone. Android applications can also be coded in Java, which is a fundamental language in programming. I will be focusing on a lightweight application for Android SDK 4.4, or the Jellybean OS of Android.

Bluetooth on Android focuses on four basic tasks. First, the Bluetooth on the smartphone must be set up with the adapter class. Second, the smartphone must find available, or paired devices. Next, the smartphone must connect to the specified device. Lastly, data transfer between the connected devices can begin. See Appendix C for sample code of this type of interaction between a smartphone application and the Bluetooth module.

Upon looking further into how to create such an application, I noticed that there are a ton of free applications on the Google Play Store that do exactly what is needed for my vision of The Tracker. However, some simple tweaking by a computer science professional should be done to the code to create the special notifications explained above, (E.g. "You have lost your keys!") instead of a simple sound notification. Also, a deeper understanding of android coding is required to create a unique and fun user interface.

The application that I decided to use for demonstration purposes is Bluetooth Notifier by ZulixSoft. The application's appearance is not ideal, but it gets the job done for proving the concept of The Tracker.

The application has three main buttons: one to turn on Bluetooth on the smartphone, one to turn off Bluetooth on the smartphone, and a third to go into the settings of the application.

Within the settings, the user can decide which actions they would like to be notified of. For example, if the user would like to be notified when the Bluetooth device disconnects, he or she can simply check that box under settings. This will alert the user with a simple notification sound when the device is out of range of the smartphone.

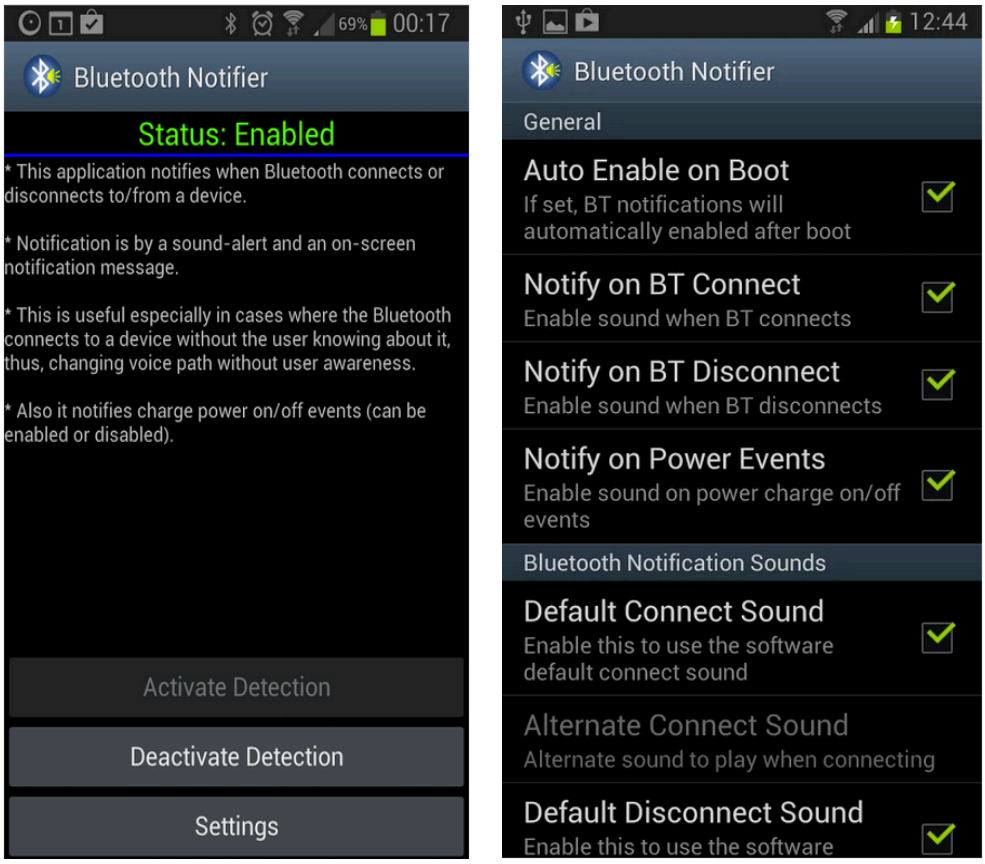

Figure 7: Bluetooth Notifier Android Application

Figure 7 reveals the user interface of the Bluetooth Notifier application utilized within this proof of concept. This user interface is representative of what The Tracker application will be able to accomplish, however it is not representative of how it should look. The application should have a clean material design that is intuitive for the user. The less text there is on the screen, without taking away from the intuitiveness of the application, the better.

Chapter 7: Future Steps

As explained throughout this report, my senior project serves only as a proof of concept. To take this design to the next level, a lot more work needs to be done. The device needs to be miniaturized and made portable. To do this, the device can be changed to utilize an Arduino Mini.

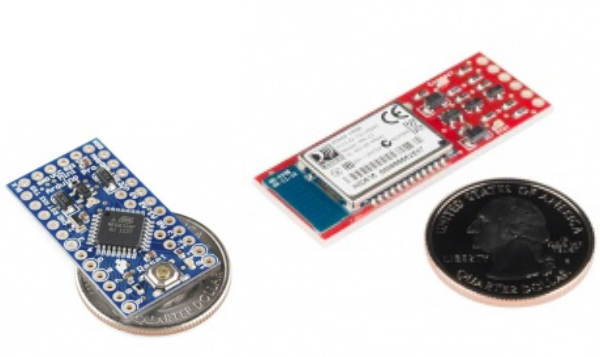

Figure 8: Small Components Enabling Design of Small Enclosure

Figure 8 shows how small the Arduino mini is in comparison to the Bluetooth Mate. This would allow the device to fit in a small enclosure, hopefully small enough to fit on a key chain. New code would have to be developed to get the Android Mini to be able to power up the Bluetooth Mate without the use of a computer. The small microcontroller would allow the design to be compact enough to fit inside a small keychain enclosure.

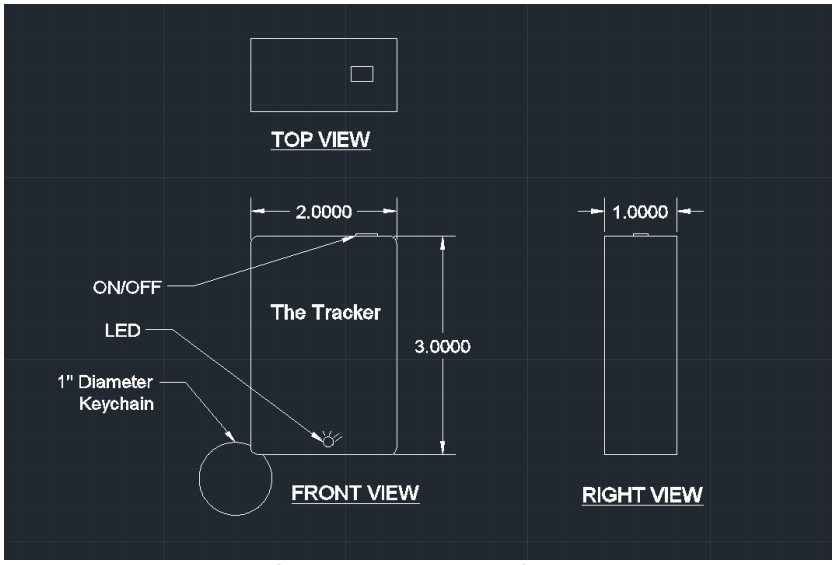

Figure 9: Concept Drawing of The Tracker

Figure 9 displays a concept drawing created with the software AutoCAD. This enclosure design will be capable of housing both the Arduino Mini, and the Bluetooth Mate. Of course, the price point of the device will need to be brought down in order to produce The Tracker at a low enough price for people willing to purchase the device. I figure that if I were to manufacture The Tracker, the price point would be dropped significantly, so I am not focusing on this aspect. Also, a cheaper Bluetooth module could be designed specifically for this project, which would allow for even more of a price drop.

A power source will also need to be designed for this project to make it portable. Within the proof of concept, the device draws power from the computer via USB. A simple battery pack can be implemented into this design to power the Arduino mini.

The smartphone application still needs to be streamlined, with customizable notifications, and a crisp user interface. All of these ideas are outside of my budget range and time allocated for this project, and should be taken up at a later date. If this project were to get investors on board, I would continue to develop The Tracker project. However for the time being, I am very happy with the way the project functions. Hopefully in the future, a portable device like The Tracker can be sold in stores near you!

Appendix A: Senior Project Analysis

Summary of Functional Requirements

The objective of this device is to be able to connect to a smartphone via Bluetooth. When the smartphone loses connection to the Bluetooth device, an application must notify the user.

Primary Constraints

A goal of this project will be to make it a small and compact device. A huge limiting factor for this goal is the power supply, since it will have the most volume. The Bluetooth module and microprocessor are both small in size, so they are not of concern. The size of the device will impact its applications and uses. Another large constraint for this project is developing an intuitive, user-friendly smartphone application that allows the user to communicate with The Tracker and will be alerted if an item is lost.

Economic

Financial capital will be present in this project. Money is needed for the R&D as well as purchasing the components for this device. There will also be Real Capital due to people investing their time and effort into this project. Also, a small amount of natural capital will go into making the components for the final device. Table IV reveals the cost estimate table.

TABLE IV
COST ESTIMATE TABLE

Part	Vendor	Quantity	Cost	Labor Cost
ATMEL Microcontroller	Sparkfun	1	$10	
Bluetooth Module	Mouser	1	$5	
Android Smartphone	Mouser	1	$30	
Breadboard	Digikey	1	$30	
Wire	OshPark	1	$30	

Costs for this project will come from buying the desired components as well as making a custom enclosure and a custom smartphone application.

If manufactured on a Commercial Basis

As of right now, I do not have a very good idea for the commercial vision of this project. However, after minor market assessments, the initial years of the product being sold might look something like this:

Estimated # of devices sold per year = 120 devices
Estimated cost of device = $10
Estimated selling price of device = $20
Estimated Profit per year = $20 x $120 = $2400
Estimated cost to operate per year = $1200

I believe the numbers above are realistic, since I do not plan on spending any money on advertising or marketing in order to get The Tracker's name out there. However this is only analysis of the first year, and I believe this technology will be subject to growth in the future, since my design is primarily focused on a proof of concept. The numbers above are contingent upon the device being developed further and finalized as a portable, and fully functional product.

Environmental

This product will have a very minimal environmental impact. Most of the components utilized will be reusable and sustainable. The only component that is to be of concern is the power supply. I do not want this to be of environmental concern to the user. For this reason, I am looking into rechargeable batteries for future design steps.

Manufacturability

Some problems that I may encounter while making this device would be making it as small and compact as possible. As I mentioned before, the size of this device is very important. Compacting all the components to fit within the desired volume constraints will be a challenge. Custom PCB's and enclosures may need to be made to allow this to happen, and this device may never reach its manufacturing stage due to a lack of funds.

Sustainability

As mentioned before, this device will use a limited amount of components so sustainability will not be an issue. I will be implementing a rechargeable battery that may need to be replaced every couple years, but as of now, I do not know the impact of the battery. I plan to have the device last for 2 years and then end the warranty.

Ethical & Health and Safety

This device would follow the IEEE COE #3. Because this device is made as a safety precaution, it will be important to make sure that the safety claims I make are true and reliable. I do not want a user depending on this device and it fails.

Social and Political

The main market that this device targets everyone with a Smartphone. The Tracker will be readily available and anyone can easily sync the device with their phone. With only a turn on/off switch on the device, using The Tracker will be a truly intuitive experience. Another option to sell this device would be to market it to universities and libraries. Universities and libraries would be an ideal market due to the purchase of bulk products. The goal of this device is to make people's lives easier. For that reason alone, it will not get us into any sort of political or social trouble. The Tracker will have privacy settings with the implementation of the sync button so that you can only connect to the device if the button is pressed.

Development

New developments for this product might include a distance sensor, to alert the user exactly how far away the object is, or a battery powered device. Also, tapping into the GPS data on a phone can make the application more intuitive. For example, a message telling the user where their keys are is sometimes more valuable than when the keys were lost. However, these developments would be aftermarket projects. This is because I need to be sure that I can make the product small enough as is before I tack on new features and developments.

Appendix B: Sample Command Mode Code for Arduino

```
int bluetoothTx = 2; // Tx - 0 pin of bluetooth mate, Arduino D2
int blueToothRx = 3; // RX-I pin of bluetooth mate, Arduino D3

SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

void setup() // allow the Bluetooth module to be connected to
{
        Serial.begin(9600); // Begin the serial monitor at 900bps
        bluetooth.begin(115200); // The Bluetooth Mate defaults to 115200bps
        bluetooth.print("$"); // Print 3 times individually
        bluetooth.print("$");
        bluetooth.print("$");
        delay(100); // Short delay, wait for the Mate to send back CMD
        bluetooth.println("U,9600,N"); // Temporarily change the baudrate to 9600
        //no parity
        bluetooth.begin(9600); // Start bluetooth serial at 9600
}

void loop()
{
        if(bluetooth.available()) // If the bluetooth sent any characters
        {
                //Send any characters the bluetooth prints to the serial monitor
                Serial.print((char)bluetooth.read());
        }
        if(Serial.avalable()) // If stuff was typed in the serial monitor
        {
                //Send any characters the Serial monitor prints to the bluetooth
                bluetooth.print((char)Serial.read());
        }
        // loop forever
}
```

Appendix C: Sample Android Bluetooth Adapter Code and Application using Java

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.bluetooth.BluetoothAdapter;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends Activity {

		@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);
		final TextView out = (TextView) findViewById(R.id.out);
		final Button turnON = (Button) findViewById(R.id.turnON);
		final Button discoverable = (Button) findViewById(R.id.discoverable);
		final Button turnOFF = (Button) findViewById(R.id.turnOFF);
		final BluetoothAdapter bluetooth = BluetoothAdapter.getDefaultAdapter();

		turnON.setOnClickListener(new View.OnClickListener() {
	public void onClick(View v) {
		if (!bluetooth.isEnabled()) {
		Toast.makeText(getApplicationContext(),
		"Turning ON Bluetooth", Toast.LENGTH_LONG);
		// Intent enableBtIntent = new
		// Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
		startActivityForResult(new Intent(
		BluetoothAdapter.ACTION_REQUEST_ENABLE), 0);
		}
		}
		});

		discoverable.setOnClickListener(new View.OnClickListener() {
		@Override
	public void onClick(View arg0) {
		if (!bluetooth.isDiscovering()) {
		Toast.makeText(getApplicationContext(),
		"MAKING YOUR DEVICE DISCOVERABLE",
		Toast.LENGTH_LONG);
		// Intent enableBtIntent = new
		// Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
		startActivityForResult(new Intent(
		BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE), 0);
		}
		}
		});
```

```
        turnOFF.setOnClickListener(new View.OnClickListener() {
        @Override
public void onClick(View arg0) {
        bluetooth.disable();
        Toast.makeText(getApplicationContext(),
        "TURNING OFF BLUETOOTH", Toast.LENGTH_LONG); }  });  }

        @Override
public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
        } }
```

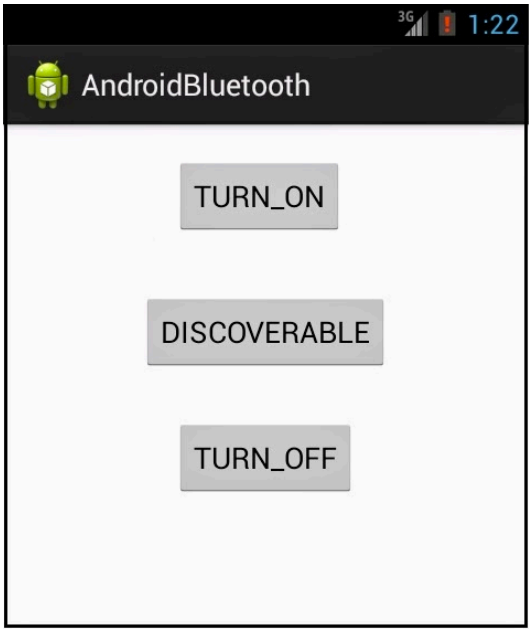

Figure 10: Screen Capture of Sample Source Code Above

Disclaimer:

The source code above does not incorporate notifications when the device is out of range of the Bluetooth device, which is the main requirement of The Tracker application. Using notifications within android involves in-depth understanding of Android Developer Tools, and coding. This is why I have opted to utilize the free application from the Google Play Store known as Bluetooth Notifier by ZulixSoft, as discussed in Chapter 6. The source code above simply serves as a proof of concept that an Android application can be created to interface with the Bluetooth module on a smartphone.

References:

1) Kline, John Wallace, and Frederick Dental. "Underwater Device with Transmitter." The Journal of the Acoustical Society of America, 121.4 (2007): 1826.

2) Gabrielson, B. Thomas, “Underwater acoustic intensity probe”
US 5392258 A, Feb 21, 1995.

3) Geller, B.; Broissier, J.M.; Capellano, V., "Equalizer for high data rate transmission in underwater communications," *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings* , vol.1, no., pp.I/302,I/306 vol.1, 13-16 Sep 1994

4) “Honeywell Pressure Sensors”, 2013, Honeywell Electronics, 2013, <http://sensing.honeywell.com/index.php?ci_id=143718>

5) Moloo, R.K.; Digumber, V.K., "Low-Cost Mobile GPS Tracking Solution," *Business Computing and Global Informatization (BCGIN), 2011 International Conference on* , vol., no., pp.516,519, 29-31 July 2011

6) Lim, D.; Anbuky, A., "Modelling and simulation of a distributed battery management system," *Industrial Electronics, 2004 IEEE International Symposium on* , vol.1, no., pp.621,626 vol. 1, 4-7 May 2004

7) Sandwith, C.; Paradis, J.; Morrison, J., "Underwater electrical cable and connector seals: Some in-house design options, commercial options, and performance/failure analyses," *OCEANS '77 Conference Record* , vol., no., pp.401,406, 17-19 Oct. 1977

8) Shahiri-Tabarestani, M.; Ganji, B.A.; Sabbaghi-Nadooshan, R., "Design and simulation of new micro-electromechanical pressure sensor for measuring intraocular pressure," *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean* , vol., no., pp.208,211, 25-28 March 2012

9) Sinkaram, C.; Rajakumar, K.; Asirvadam, V., "Modeling battery management system using the lithium-ion battery," *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on* , vol., no., pp.50,55, 23-25 Nov. 2012

10) Yu-Hung Hsiao; Min-Chih Huang; Chau-Chang Wang, "Development of MSP430-Based Underwater Acoustic Recorder with Multi-MCU Framework," *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on* , vol., no., pp.101,106, 17-20 April 2007