

THE FIDO FINDER APPLICATION USER GUIDE

Grayson Goering and Sarah Gillogly

TABLE OF CONTENTS

1. Materials Needed.....	3
2. Application Overview.....	6
3. Walkthrough.....	6
4. Conclusion.....	7

1. MATERIALS NEEDED

This application requires other physical hardware for it to be fully functional. Some soldering may be required to make these items function, a kit may be purchased here:

<https://tinyurl.com/36z2pdtr>. Here is a list of the items needed to have this app working properly:

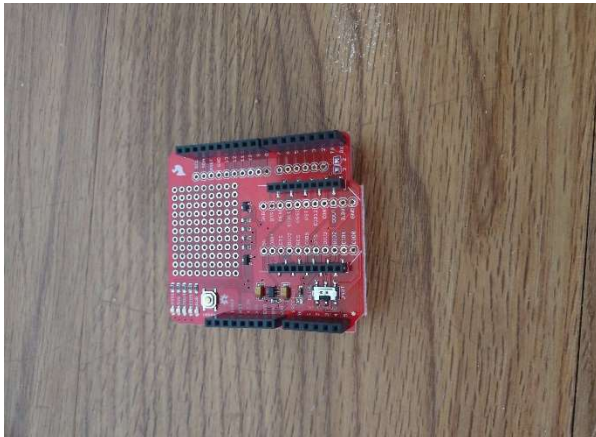
- a. One Arduino Uno Board



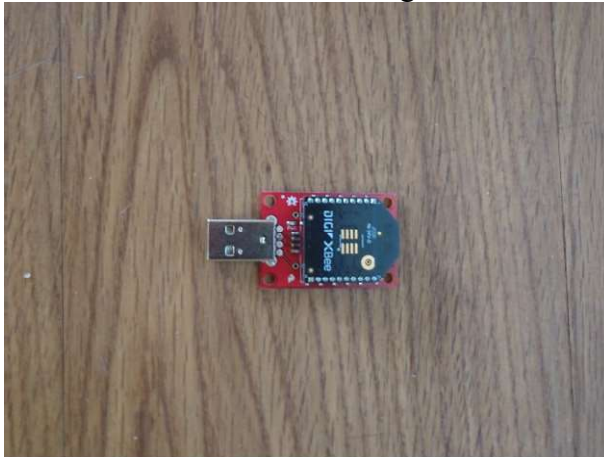
- b. One Arduino GPS Shield



- c. One Arduino XBee Radio Shield



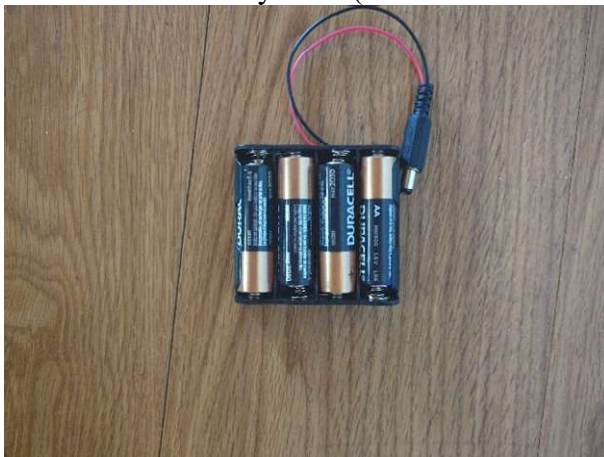
- d. One Arduino XBee Radio Dongle



- e. Two XBee radios



- f. One Arduino Battery Pack (4 double A batteries needed)



- g. USB to USB-C adapter (This depends on the device that you own. Get an adapter that adapts from USB to what kind of charging port your phone has built in.)



These items may be purchased from this link: <https://www.sparkfun.com/products/11021>. If

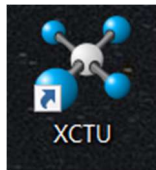
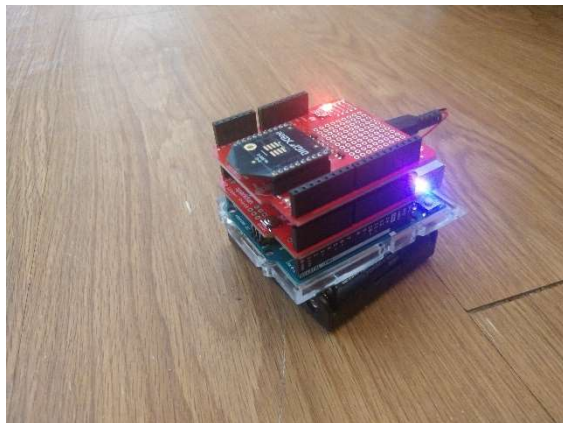


Figure 1.
An XCTU
shortcut icon.

you have these items, a few more steps must be taken. The XBee radios must be activated to broadcast and receive from the same signal. A program called XCTU should be able to help with setting up the radios properly. Here is a tutorial that shows how to do this: <https://tinyurl.com/67hvmkzn>. This tutorial shows how to set the address for each radio. If the addresses for broadcasting are the same, the radios should be able to work in one way communication for the purpose of this app. Below are images that indicate how these pieces should be put together for this app.



The Arduino Uno board needs to be connected to both Arduino Shields (GPS and XBee). An XBee radio needs to be connected to the Arduino XBee Radio Shield and the other radio needs to be connected to the Arduino XBee Radio Dongle. The radio in the shield will be broadcasting to the radio on the dongle. The dongle attaches to the USB adapter to utilize your phone as a power source to receive a broadcast. The Arduino Uno board connects with the Arduino Battery Pack for a power source. The final set up step requires

Arduino code to be uploaded into the Arduino Uno Board. This code is linked in the following GitHub repository <https://github.com/sgillogly/finalPetTrackerApp> in the file titled ArduinoCode. Once you have these materials put together and the Arduino code is uploaded into the Arduino Uno Board, make sure that the XBee radios are getting and receiving signals. This is indicated through the Rx (receiving) and the Tx (transmitting) LEDs that are on both the Arduino XBee Radio Dongle and the Arduino XBee Radio Shield. The XBee radio connected to the shield should have the Tx LED on to indicate that it is transmitting and the XBee radio that is connected to the Arduino XBee Radio Dongle should have the Rx LED on to indicate that it is receiving information from the other XBee Radio. If these lights are not on, the application will

not work. Take the time to make sure that the radios are properly set up before attempting to use the application.

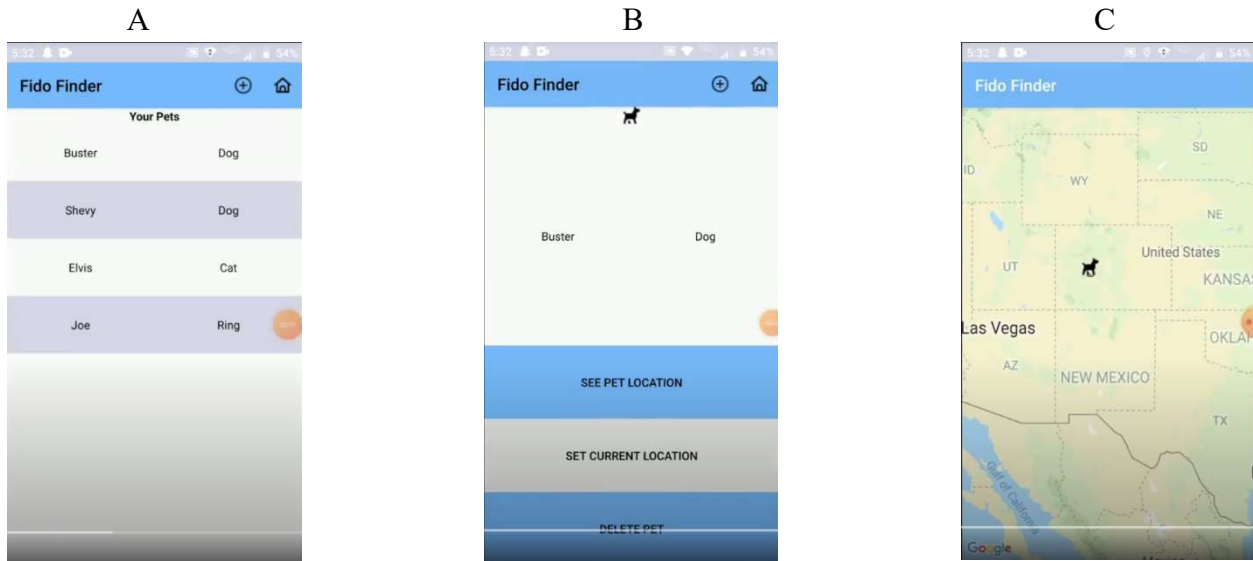
The application will run without these materials, but the location services will be of no use. In theory, the large Arduino Uno Board, with everything attached to it, would be what the user's pet would wear should they escape. This application is a Computer Science Capstone Final Project and thus prototypes are needed to be able to make the application function as it should as we are not Electrical Engineers. If the user has these pieces, sets them up correctly, and uploads the Arduino code into their Arduino Uno Board, their Fido Finder application will work properly.

2. APPLICATION OVERVIEW

The Fido Finder application is meant to help users find their pets if they have run away. In execution, this would mean that the Arduino hardware pictured above would be strapped to your pet. This is not a practical option for either yourself or the pet, so this project is mainly a prototype.

With all the pieces assembled and the app downloaded, a user will be able to view the location of the assembled hardware as well as their home location (if it is set) and the current location of their phone.

3. WALKTHROUGH



The above images provide a walkthrough of the Fido Finder application. Image A is the home screen of the app where all your pets may be viewed. The hardware for this application is very limiting, so while you can add multiple pets to your home screen, the radio that is connected to your phone is only connected to the other radio which is attached to the rest of the hardware. If you wanted a certain pet's location to be seen, the hardware would need to be on that pet specifically.

Image B is the detailed view of a user's pet. The first button allows for the user to open their pet's location as seen in image C. The second button, "set current location," allows the user to set their current location as the home location for this pet. If this is set, then image C will also display a small home icon at the location that was stored. This can be updated at any time. The final button is the "delete pet" button which allows the user to remove this pet from their app.

4. CONCLUSION

Thank you for taking the time to read this document and looking at the Fido Finder project. This project is the Senior Capstone Project for Sarah Gillogly and Grayson Goering at Western Colorado University, class of 2021.