Toddler Tracker
EECS 129B
March 17, 2008

Last Name	First Name
Valentine	Megan
Ingel	Francis

0. Table of Contents

1. Abstract

The goal for our ultimate product is to have a system that can monitor a child using a danger beacon to set an area as off limits. To ensure child safety, a child cannot step within a danger radius set on the danger beacon. Once the child does step within that radius an alarm with be set off on the danger beacon to alert any adults within the area. The danger radius also contains a peril radius which triggers a faster alarm.

To prevent the alarm from going off if an adult is in the room with the child, there is a button that will turn the system off for a short period of time. In order to ensure that the button is protected from the child, it must first be unlocked by the execution of a step sequence outside of the danger radius before. After being activated, pressing the button increments a timer that will keep the danger beacon off until it expires. If the button is pressed again before the time is up, the duration Is incremented with each additional press. Additionally the device can be prematurely turned on by holding the button down.

2. Project Overview (High level)

Purpose: The purpose of this project is to produce a child monitoring device that can aide in ensuring the safety of a child in the home. We wanted to make a product that provides versatile and portable protection for children that is not too complicated to use.

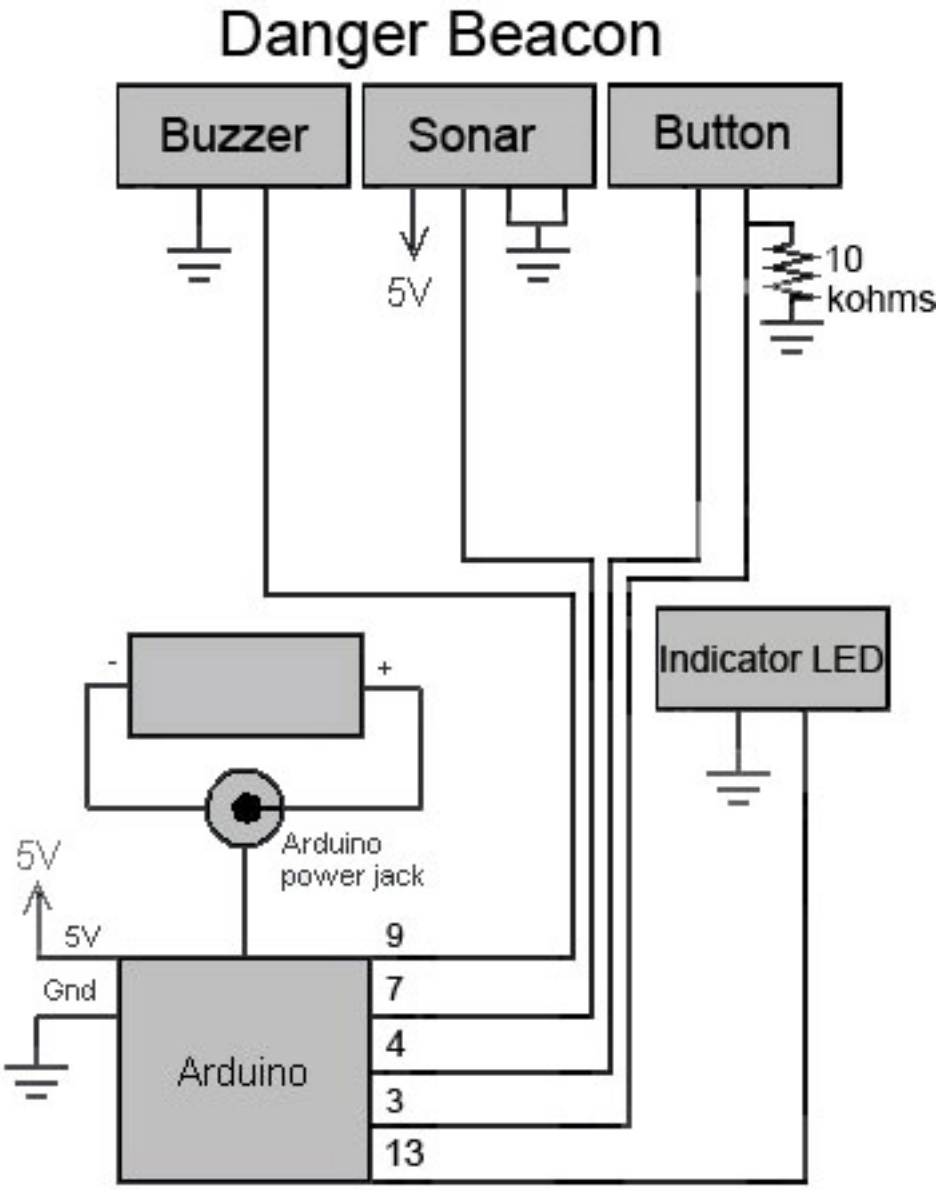
Background: Child safety is a primary concern for many people considering all of the dangerous things that are located in the home. From the standpoint of someone who has had the experience of babysitting, an extra set of protection is always useful. Because there are no products like it on the market, we had to really think of what functions we wanted to implement. It was important to make sure that the device was versatile enough to be useful in many different situations and for many different people. Though the final project is entitled Toddler Tracker, the original idea came from the desire to make the home safer for the elderly with dementia by providing aide to their care takers. This product can be marketed in so many different ways and helps to fill a need in two completely different environments. Furthermore the usefulness and practicality of the device add to its appeal. The planning that went into the development of the concept provides uniqueness to the project and was a definite challenge. There was very little example to follow and was really innovating. Also, the compact single unit makes it less expensive and easier to manage.

Features: The sonar returns the distance of any object within its range. Based on this value, the buzzer plays two sets of tones if anything is detected within the Danger or Peril Radius from the Danger Beacon. Also, the sonar records the previously detected distance and uses this to recognize when a set of steps is done to unlock the Interrupt Button. The Interrupt Button is used to temporarily shut off the Danger beacon based on how many times it is pressed. Holding the button down for at least one second will resume normal operation.

System Configuration: The system will require an Arduino microcontroller, small breadboard, a proximity detector, a status indicator, and a button.

3. Project Components

- **Electrical**



- Buzzer: plays the alarm with a signal from the Arduino
- Sonar: provides the distance of the closest object to the Arduino
- Button: allows user input to the Arduino
- LED: blinks to indicate activation of the sonar

- **Software (Programs)**

The sonar performs periodic searches for objects within either of its two designated radii by using a counter to measure the echo time. Based on this echo time, the approximate distance of the detected object can be calculated using ticks per inch ratio that was experimentally computed. The Danger Radius is defined at 6 feet, where an alert is made that something is approaching the Danger Beacon. The Peril Radius is defined at half the distance of the Danger Radius and causes a more intense alert that an object is too close to the Danger Beacon. The alerts are done through a buzzer with two variations. When within the Danger Radius, the buzzer outputs a sequence of tones, but when within the Peril Radius, the sequence of tones is played at a higher frequency. The Arduino keeps track of the previously recorded distance and watches for the initiation of a step sequence that will unlock the Danger Beacon. By standing within one feet outside the Danger Radius, take a small step backward with the double blink of the LED and take a small step forward with the following triple blink of the LED. The LED will blink four times and play the peril alert with the buzzer to indicate that the device will be unlocked for the next minute, allowing use of the Override Button.

The Interrupt Button is a part of the Danger Beacon, but acts separately in that it serves to interrupt normal operation. One press of the button initiates a temporary shutoff mode. Any subsequent presses of the button within a designated one second window adds an additional preset unit of shutoff time. During this shutoff mode, holding down the button for one second will prematurely terminate the shutoff mode and resume normal operation. Also during shutoff mode, any other presses of the button adds an additional preset unit of shutoff time.

Key Algorithms

```
setup()
  set LED, sonar, and speaker to OUTPUT
  set pin 0 to call button() when the value CHANGEs
  begin Serial at 9600

loop()
  check the lock
  if (button is active)
        if (shutoff time is over)
              set button to idle
  if (button is waiting)
        if (done waiting for more input)
              set button to active
              set the shutoff timer to (# of presses * the unit for the shutoff timer)
  if (button is idle)
        set sonar to OUTPUT
        buzzerOn(activateSonar())
```

buzzerOn(state)
```
  if (state is in peril)
        play quickly
  else if (state is in danger)
        play normally
```

activateSonar()
```
  turn the sonar HIGH
  send a LOW HIGH LOW pulse
  set sonar to INPUT
  wait for a HIGH
  while (still HIGH)
        count the echo pulse time
  turn the sonar LOW
  if (the button is locked and step sequence is detected)
        unlock the button for a set period of time
  if (the echo distance is within the perilRadius)
        the state is in peril
  else if (the echo distance is within the dangerRadius)
        the state is in danger
  else
        the state is safe
```

button interrupt()
```
  if (the button is locked)
        do nothing
  if (the button is being pressed)
        record the start time
  else
        calculate how long the button was pressed
  if (button is idle)
        set the button to wait for input
  else if (button is waiting for input)
        count how many times the button is pressed
  else if (button is active)
        if (the button was held for at least a second)
              clear the shutoff timer and set button to idle
        else
              increment the shutoff timer by one unit
```

4. System Test Plan

Test Number	Description of Set-up	Input or Stimulus	Expected Behavior
1	Place Danger Beacon on elevated surface (table, counter, etc.) in an open room towards an entrance. For this device to work placement is crucial it needs to be facing in the direction where there is direct access to the object.	A person walks toward the Danger Beacon but remains outside of the Danger Radius	Nothing happens.
2	Same set up for Danger Beacon in previous test case.	A person walks toward the Danger Beacon and enters the Danger Radius.	After entering the Danger Radius, the sonar detects the object and plays a slow sequence of beeps.
3	Same set up for Danger Beacon in previous test case.	A person walks toward the Danger Beacon and enters the Peril Radius.	After entering the Peril Radius, the sonar detects the object and plays a fast sequence of beeps.
4	Same set up for Danger Beacon in previous test case.	A person walks toward the Danger Beacon and enters the Danger Radius then walks back.	After entering the Danger Radius, the sonar detects the object and plays a slow sequence of beeps. Once they leave, the buzzer is no longer activated.
5	Same set up for Danger Beacon in previous test case.	Stand within one foot outside of the Danger Radius and take a step backward after a single blink.	Nothing happens.
6	Same set up for Danger Beacon in previous test case.	Stand within one foot outside of the Danger Radius and take a step backward after a double blink.	The device blinks three times.

7	Same set up for Danger Beacon in previous test case.	Continuing from #6, after the triple blink, do anything except taking a step forward.	The device blinks once.
8	Same set up for Danger Beacon in previous test case.	Continuing from #6, after the triple blink, take a step forward.	The device blinks four times and plays the fast sequence of beeps as if someone was in the Peril Radius. This then disables the lock on the button.
9	Same set up for Danger Beacon in previous test case.	Press the button once	Nothing happens because the device is locked.
10	Same set up for Danger Beacon in previous test case.	Continuing from #8, after the lock is released, press the button once	The Danger Beacons waits for a second for any further input then shuts off for a period of time designated within the program.
11	Same set up for Danger Beacon in previous test case.	Continuing from #8, after the lock is released, press the button once, then a second time before a second passes	The Danger Beacon waits for a second after the second press of the button then shuts of for twice the time designated within the program.
12	Same set up for Danger Beacon in previous test case.	Continuing from #8, after the lock is released, press the button once. When the Danger Beacon is in shutoff mode, press the button again.	The Danger Beacon adds another preset unit of shutoff time

13	Same set up for Danger Beacon in previous test case.	Continuing from #8, after the lock is released, press the button once. When the Danger Beacon is in shutoff mode, hold the button down for a second.	The Danger Beacon immediately exits shutoff mode and resumes normal operation
14	Same set up for Danger Beacon in previous test case.	Continuing from #8, after the lock is released, wait for one minute and then press the button once	Nothing happens because the device is locked again.

5. Project timeline

Milestone	Date
Choose the Project	Wed 11/14
Ordered Parts	Winter Break
Got the Sonar Device to Work	Weekend After Week 3 Winter Quarter
Changed from Vista to an XP System	Weekend After Week 8
Got the Button Timer to Work	Weekend After Week 8 (Needed XP)
Gave up on the XBee	Week 10 Monday
Got Infrared to Work	Week 10 Tuesday
Infrared Stopped Working (Pototransistor)	Week 10 Thursday
Programmed the Unlock Button Feature Using Steps	Week 10 Wednesday Night

6. Division of work

Most of the work on the project was done together because this was a learning experience for both of us so using each other's input as we worked was really valuable. Francis headed up the programming and Megan focused more on the electrical work including soldering and the ordering of parts. We both took responsibility for debugging the circuits in the lab and we tested the projects together.

7. Cost

Part	Cost Per Unit	Number of Units	Total Cost
Arduino	$34.95	1	$34.95
Devantech SRF05 (Sonar Device)	$29.50	1	$29.50
Buzzer	$1.55	1	$1.55
Button	$2.49	1	$2.49
Mini Breadboard	$4.99	1	$4.99
Reel of Yellow Wire	$5.49	1	$5.49
Reel of Red Wire	$5.49	1	$5.49
Reel of Black Wire	$5.49	1	$5.49
Rosin Core Solder	$3.29	1	$3.29
10Kohm Resistor	$1.49 (2 Pack)	1	$1.49
Red LED	$0.06	1	$0.06
Heat Shrink Tubing	$2.19	1	$2.19
AC/DC Adapter	$8.95	1	$8.95

Total cost of project: $106.38

8. Conclusion

We encountered many problems implementing communication between the ZigBee wireless chips. When we went into lab we were able to debug the circuit to make sure that we were getting the correct voltages throughout the circuit. We were never able to get a signal from the ZigBee. Suspecting that the ZigBees were fried, during tenth week, we had to give up on our original idea and try to think of a different course of action. When we came up with the idea to use infrared as a way to unlock the button function we seemed to be back on track until our phototransistor stopped working. We came up with the step sequence to solve our problem. Having these issues with our project really challenged us and made it so that we had to come up with quick solutions and really made us exercise our creativity.

This course definitely helped us learn a lot more about microcontrollers. The entire project was something different than anything that we had done in previous courses, and it was great to use both electrical and programming skills on the same project.