ECE 445

Anti-lost/theft Alarming System for Personal Belongings

Final Paper

Project 24

Wenhao Li – Yunchi Sun – Xiying Wang

TA: Igor Fedorov

Contents

1. Introduction

1.1 General Introduction

Many precious personal belongings, either by theft or carelessness, are easily lost or cause serious damage. According to a report by TechCrunch: US citizens, on average, lost one smartphone annually, which caused 30 billion dollar loss of money in 2012. Not to mention the loss of wallet, which may cause the loss of your driving license, cash and credit cards.

Our idea is to use wireless communication technology to remarkably decrease the loss rate of those important items you carry. Since those personal belongings (i.e. wallet, cell-phones, keychain) are supposed to be very close to you, we would pair them up (use transmitters and receiver) with a portable base device that could be easily hooked up/carried in your jacket/coats. This device will detect if any of the personal belongings are too far away from you (for example more than 2-5 meters) and send out an alarm indicating this item is under potential risk of loss. The buzzer on the tags side will beep when it is out of range, therefore, the user is able to trace the losing item according to the sound.

The listed blocks and modules will be explained in greater depth in the next section.

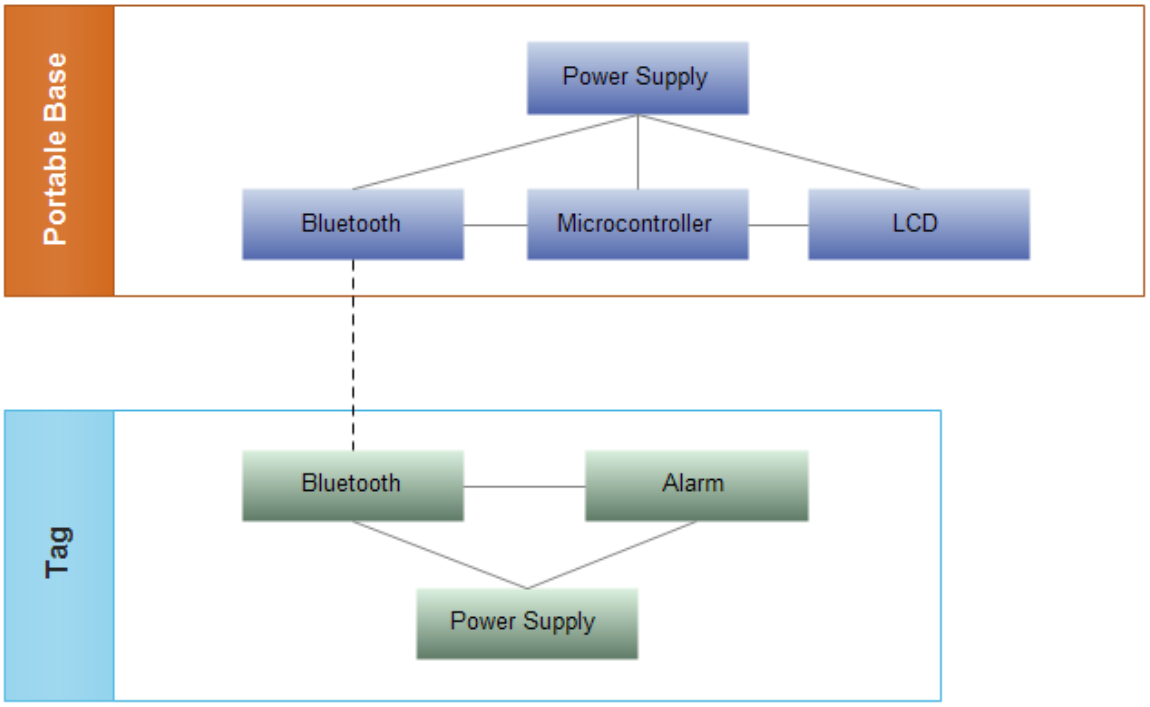


Figure 1. Top-level Block Diagram

1.2 Blocks and Modules

Microcontroller Module

Microcontroller Module will retrieve data from Bluetooth Module, once a tag is out of range and under potential risk of lost, it will trigger the alarm by commanding the Bluetooth Module sending out the signal. Arduino UNO microcontroller will be used, it will interfere with the Bluetooth as well as interfere with LCD screen.

Power Supply

On Portable base and tags, CR2330 Lithium coin batteries, operating 3.2V, will be used to power up Bluetooth chips. In addition, a 9V battery from Eveready will be used as power source for LCD screen and microcontroller on the master-end.

Communication

The wireless communication between two ends is achieved by Bluetooth 4.0 technology. The Bluegiga ble112 chip will be used and programmed to make communication. The configuration and connection methods will be explained in details under design portion.

Display Module

LCD is working under direction of the microcontroller. It will display the information needed to users. The ACM1602A SERIES LCD screen will be used. It will interfere with microcontroller via SPI and be powered by power supply module with 9V DC battery.

Buzzer

A FY14. 3-18Vdc mini-piezo buzzer will be used which will emits a medium, high-pitched tone when energized. It mainly provides a way for user to trace the losing item when applicable. The buzzer is powered up directly by the Bluetooth chip on tag side.

Push Button and LED

A small push button will be used on the portable base to turn off the notification displayed on the LCD screen. Also a LED will be used on the portable base, indicating user when an item is under potential loss. The LED can also be controlled by the push button.

1.3 Functionality

We intend to contain following features in our project. The portable base is able to track multiple items at the same time and also distinguish each from others. The LCD screen is able to indicate the item information when it is under potential risk of loss. The LED flashes at the same moment of screen displaying information. On the tag side, the buzzer is triggered automatically when the item is out of range. The most important function is that the Bluetooth chips on both sides are able to communicate with each other by sending data. The Bluetooth on the master end should also send data to the microcontroller to notify certain item is out of range and microcontroller controls the LCD and LED. Another feature on the master end is to manually turn off the LED and LCD notification by pushing the button.

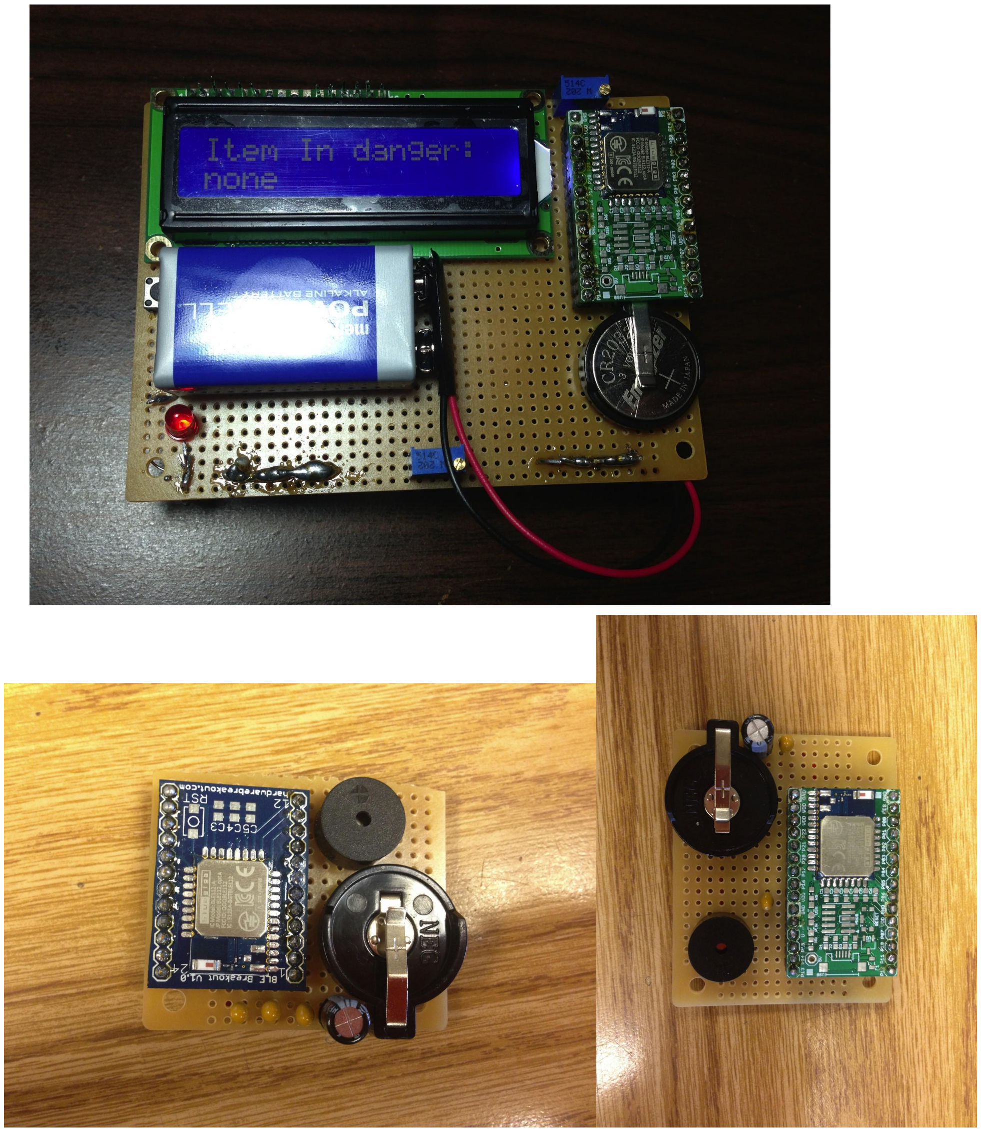


Figure 2. Portable Base and Tags

2. Design

2.1 Portable Base (Master End)

2.1.1 Power Supply

Power Supply Module is consisted with two parts, 9V battery from Eveready and a CR2032 Lithium coin cell battery. The 9V battery is mainly providing voltage for LCD screen and microcontroller. The coin cell battery is only used for Bluetooth chip in order to have enough voltage to transmit data with slave end Bluetooth chip. Since the LCD consumes large power, we chose to use 9V battery because of its cheap price and constant voltage regulating. The LED and push button will be directly powered by LCD screen.

2.1.2 Microcontroller

Arduino UNO was selected as our microcontroller. It will be connected with communication module so it is able to receive a serial input code containing the missing item type, after decoding the input, it will send instructions to control the LCD module displaying the missing item name. The microcontroller module connected to a LED light and a button, which could potentially alarm the user via bright the LED light when something is missing. User is able to disalarm using the button.

2.1.3 Communication

Bluegiga BLE112 chip was used in the communication module. The chip was programmed to advertising all the time to let tag detect the signal strength. If the tag detect that the RSSI is lower than the setup reference, the tag will make connection to the Portable base communication module. This module will receive the identification information from the tag wirelessly and send the information to the serial input of the microcontroller.

2.1.4 LCD

We have used ACM1602A LCD screen on the portable base. The main function is to display the item information for users. It is controlled by the Arduino UNO and all the item information is programmed in the microcontroller. When one item is out of range, the Bluetooth chip receives the data from slave end and sends signal to microcontroller. Then the microcontroller will call the display function for LCD screen and indicating that certain item is in danger. The information will be displaying all the time until the user brings the losing item back and presses the push button. In this case, the push button will turn off the alert on the display and the screen goes back to normal functionality. The LCD is mainly powered by the 9V battery.

2.2 Tag (Slave End)

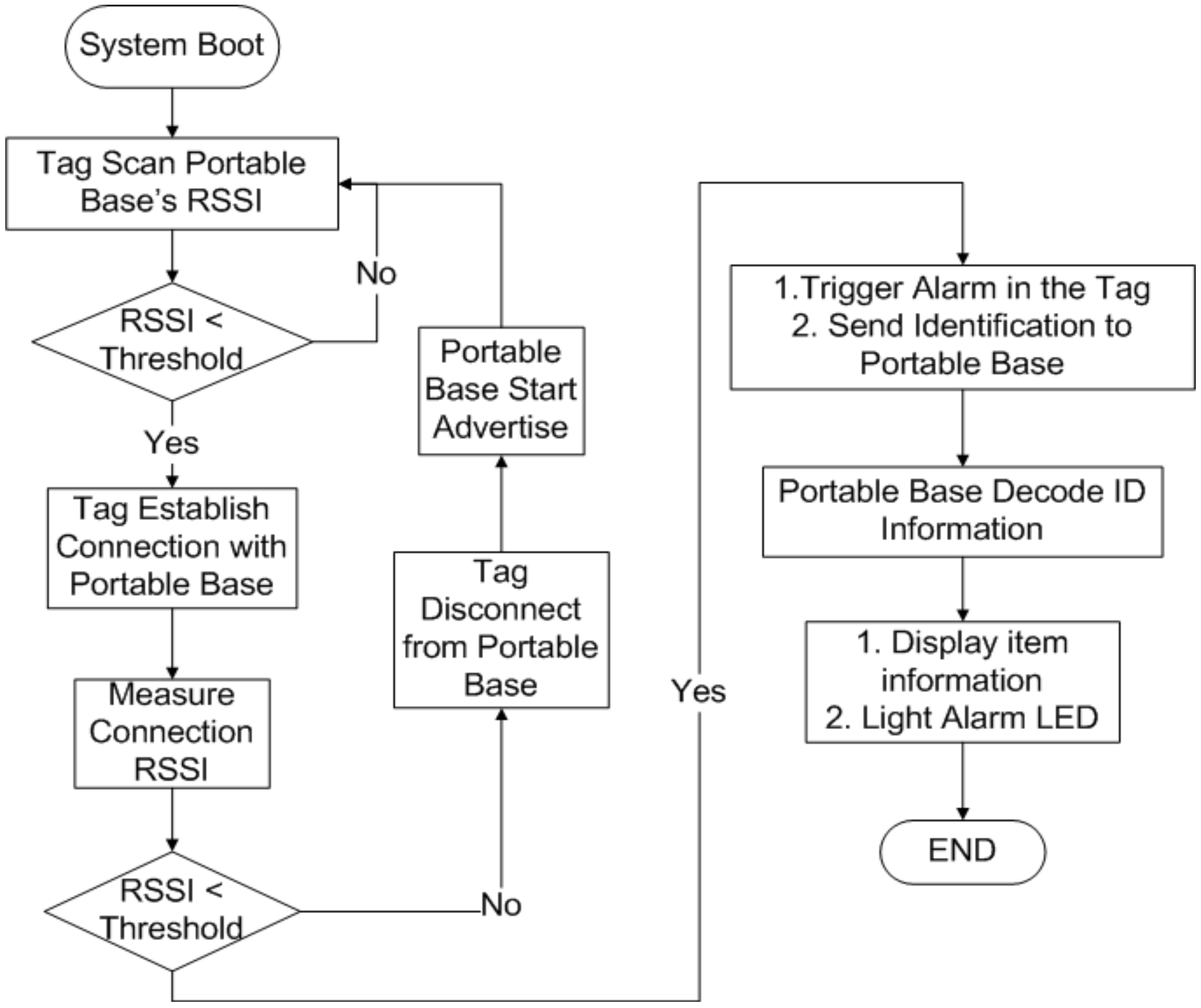

System Boot
Tag Scan Portable Base's RSSI
RSSI < Threshold
No
Yes
Tag Establish Connection with Portable Base
Measure Connection RSSI
RSSI < Threshold
No
Tag Disconnect from Portable Base
Portable Base Start Advertise
Yes
1.Trigger Alarm in the Tag
2. Send Identification to Portable Base
Portable Base Decode ID Information
1. Display item information
2. Light Alarm LED
END

Figure 3. System Flow

2.2.1 Power Supply

Power Supply Module (Slave-End) will power up the Bluetooth chip using CR2032 Lithium coin batteries with rated output voltage at 3.2V. TI's TPS62730 chip, an ultra low power DC/DC converter with by-pass mode will be used to reduce the current consumption of Bluetooth chip during transmission nominally by ~20%.

2.2.2 Communication Module

Bluegiga BLE112 chip was used in the communication module. The chip provided the Received Signal Strength Indication (RSSI) command to acquire the received signal strength. The communication module scanned the RSSI of the Portable base Bluetooth. By comparing the RSSI values with the setup reference value to determine if the tag is within certain range from the Portable base. When the RSSI from the Portable base is less than the reference value, it will establish a connection to the portable base first, and then, acquire the RSSI again. If the RSSI is still less than the reference value, the chip will trigger the alarm and send its identification to the Portable base. If the RSSI is larger than the reference, it will disconnect from the portable base and start scan again.

2.2.3. Alarm Module

The buzzer was directly powered by the BLE 112 chip. It is connected to the output pin P0_7 on the Bluetooth chip. The programming function in the Bluetooth chip asks the pin to output high when the RSSI value is lower than the threshold, meaning that the item is out of safe range. Then the buzzer will start to beep since it receives high input. The Bluetooth itself consumes very little current, so that we directly power the buzzer using Bluetooth. The buzzer itself has low resistance, which makes power enough to maintain high pitch sound.

3. Requirements and Verifications

3.1 Portable base

The first requirement for the portable base including supplying correct voltages to Arduino UNO(around 5V) , display(around 5V) as well as the Bluetooth chip.
To test the power supply, we connect multi-meter in series to the battery to make sure the voltage is within the range

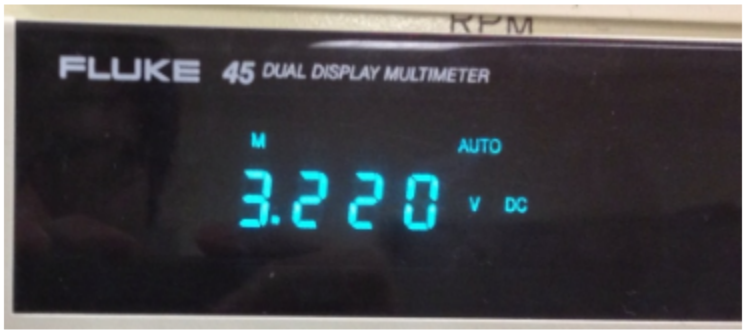
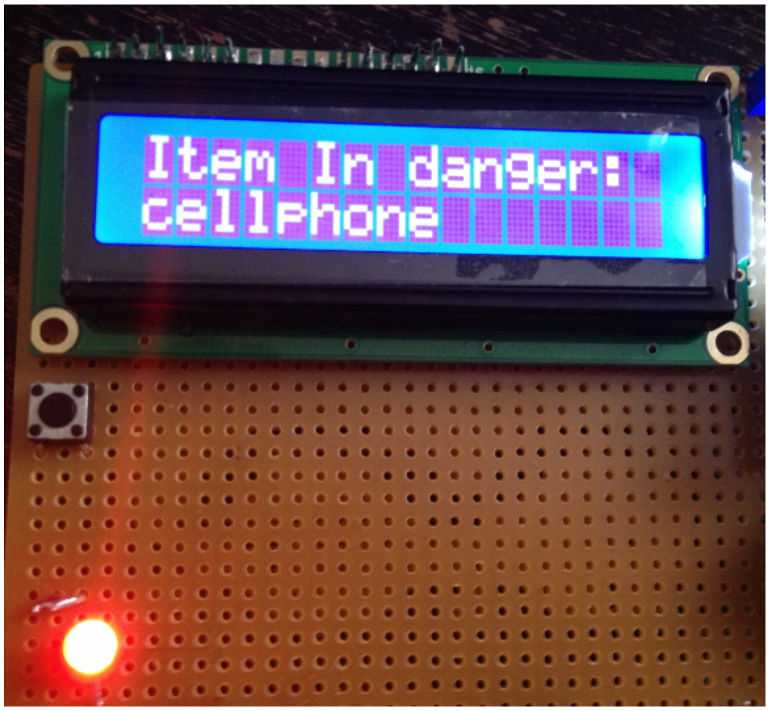


Figure 4: Multimeter Reader

Another requirement is the base must be able to identify the signal transmitted by the tag and display the identity of the item in danger.
To verify this function is working correctly, we put one tag out of range, check if the display indicated the right name assigned to the missing tag and check if the LED was light up as

programmed to. Then we perform the same test on another tag and see if another correct name if indicated by the display.



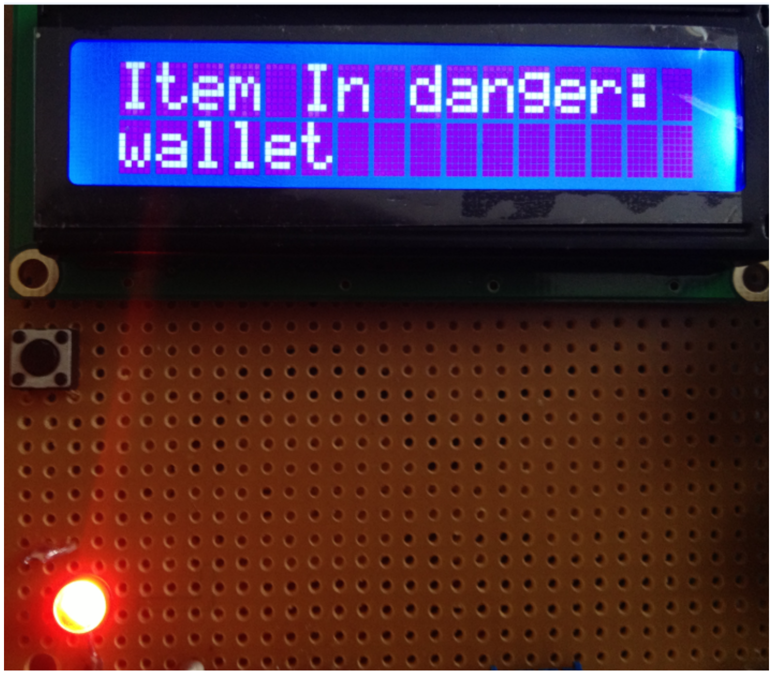


Figure 5&6: LCD indicate what item is losing

Another requirement is the base must be visible to each tag. We will introduce how we make connections and everything about wireless communication in next section. Test and verifications will be on next section as well

3.2 Tag

The requirement for the tag is to acquire the RSSI values from the Portable base every three-second and according to the RSSI to determine if the tag is out of range. If the tag is not within the safety range, the Bluetooth chip will trigger the alarm and send its identification to the portable base.

The test we performed to get the relationship between the distance and the RSSI was set the tag Bluetooth in scan mode and connect it to the PC via USB, then place the Portable base 0.5 m away from the tag, monitored the RSSI of the Portable base Bluetooth on the PC. And do this experiment several times in different directions. After determine a reasonable threshold RSSI for 0.5 m, we did the test for 1.5 m again. The following plots were made by two sets of data, one from 0.5 m and the other from 1.5 m.

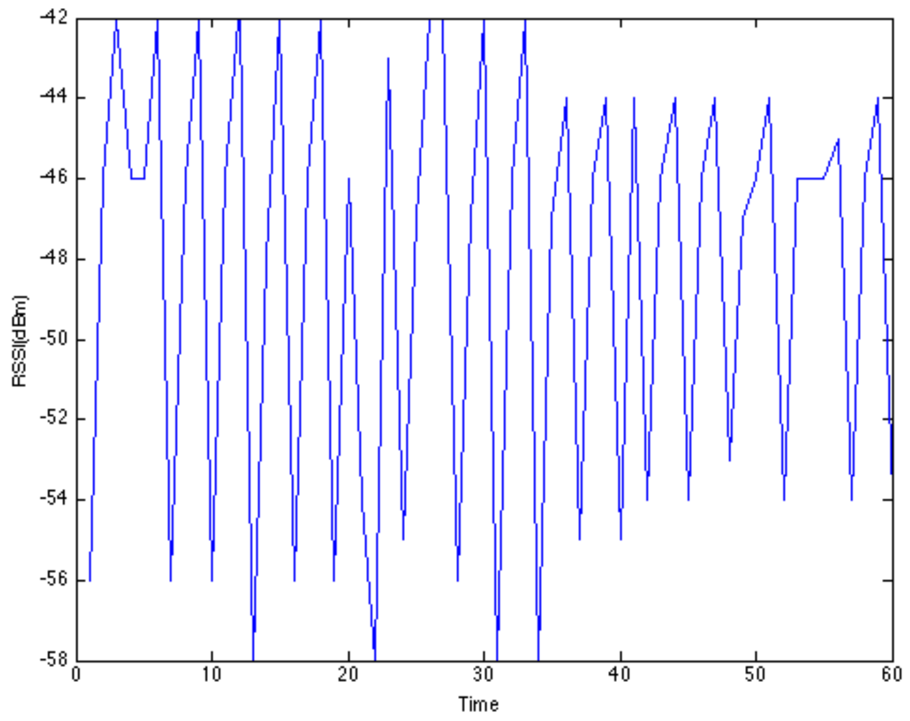


Figure 7: RSSI at 0.5 m

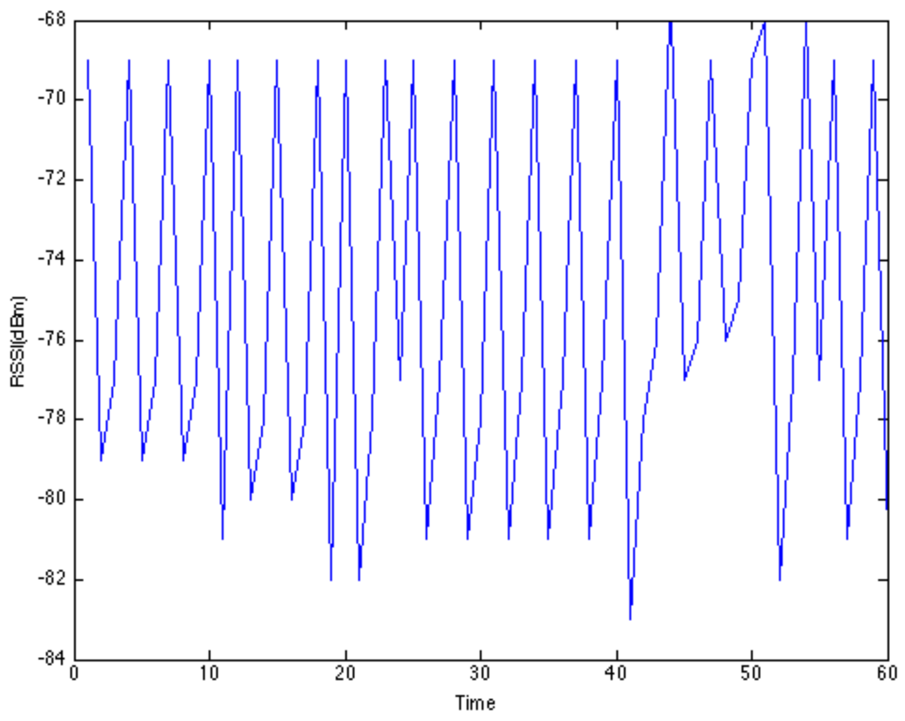


Figure 8: RSSI at 1.5 m

As the RSSI was noisy, we applied a digital low pass filter to the signal to reduce the chance of false alarm. The following is the pseudo code we used to implement the low pass filter:

```
Start:
        Input D
        S = (f * D) + ((1 - f) * P)
        Output S
        P = S
        goto Start
End
```

where	S:	filtered value
	f:	Filter factor
	D:	Raw data input
	P:	Previous value of S

After several experiment we chose the filter factor to be 0.2 and the following plots are the RSSI after the low pass filter.

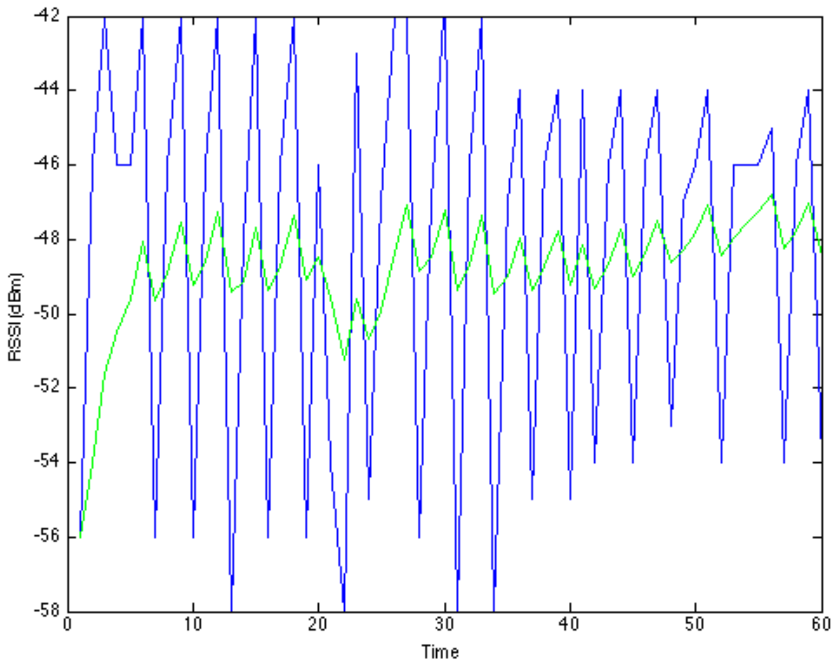


Figure 9: Filtered RSSI at 0.5 m

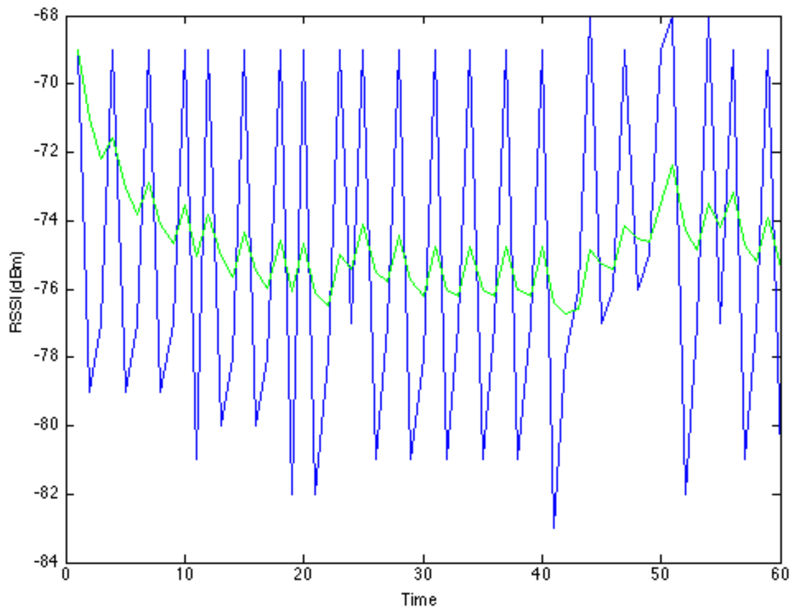


Figure 10: Filtered RSSI at 1.5 m

The black line is the raw RSSI input and the green line is the RSSI after the filter. We can see that the signal become more smooth. Based on these test, the threshold RSSI for 0.5 m is -52 dBm and for 1.5 m is -77 dBm.

The following plot was created by the real test profile while testing, first we place the Portable base right beside the tag, after several second, the Portable base was moved 2.5 m away from the tag and then move back beside the tag.

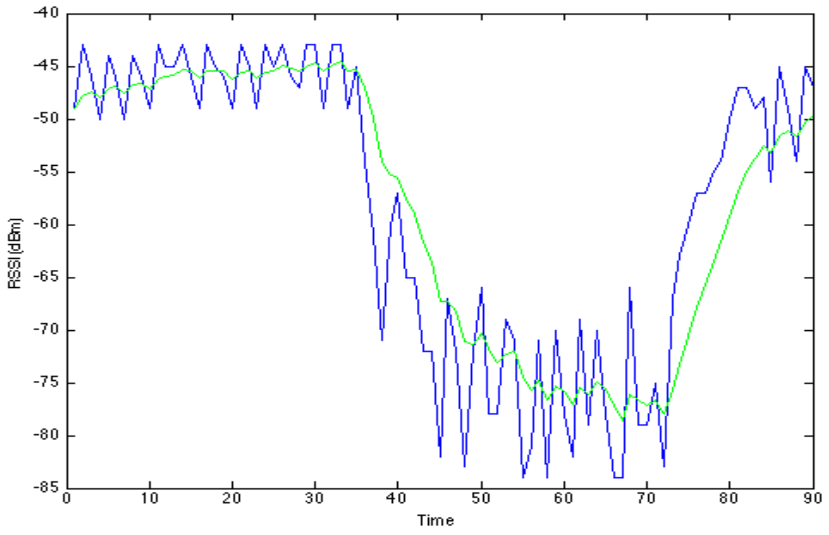


Figure 11: Real RSSI Test Plot

From the plot, the RSSI was around -45 to -50 dBm when the Portable base was placed beside the tag. After the base was move to 2.5 m away, the RSSI reduced to in the range of -73 to -78 dBm.

The other test was based on the RSSI test, we have set the threshold RSSI for one tag to be -52 dBm (0.5 m). We verified that the alarm could be triggered and the Bluetooth chip could send out the correct ID information to the Portable base.

3.3 Failed Requirements

When designing the system, we would like to use the recommended TI TPS62730 buck converter to improve the power efficiency. However, there was a big issue when soldering the chip. The TPS62730 was a dry package chip which has a dimension of 1.5mm by 1.05mm. The six surface mounted pin were needed to solder onto the PCB. This requires professional tools to do the work and we are not able to solder the converter to our PCB in the lab. The following picture shows how small the converter is compare to the tip of a pen. This was the only failed requirement we had. We are short of time in the end of the period and did not happen to research more and choose a different power converter for the Bluetooth chip. However, this failed requirement does not affect the functionality of our project which is tracking items correctly. If the time is allowable, we would have solved this problem by replacing another possible power converter to improve the efficiency of the system.

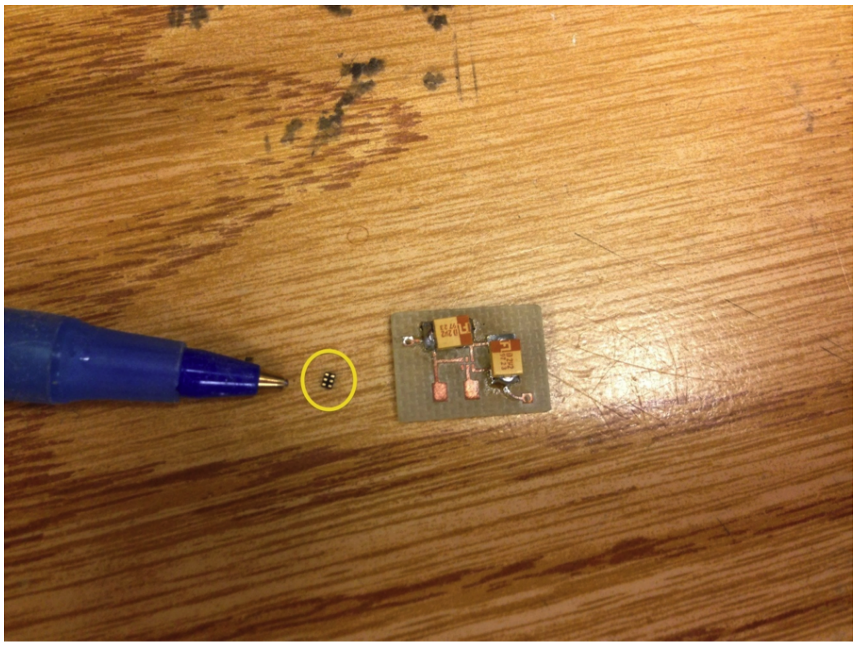


Figure 12: Power Converter and PCB

4. Cost

4.1 Parts

Item	Price/Unit	Quantity	Cost ($)
PCB	$25	2	$50
LED (HLMP3301)	$0.15	10	$1.5
Microcontroller (Arduino UNO)	$30	1	$30
Bluetooth chip (Bluegiga BLE112)	$17.45	5	$87.25
Buzzer (FY-14)	$1.81	5	$9.05
LCD Display (ACM1602A)	$34	1	$34
Coin Cell Battery (CR2032)	$1.97	10	$19.7
9V Battery (EN22)	$1.34	3	$4.02
Converter (TI TPS62730)	$2.02	5	10.1
Total			$245.62

4.2 Labor

Name	**Hourly Rate**	**Total Hours Invested**	**Total = Hourly Rate x 2.5 x Total Hours Invested**
Wenhao Li	$35	180	$15750
Yunchi Sun	$35	180	$15750
Xiying Wang	$35	180	$15750
Total		540	$47250

Grand Total

Section	**Total**
Labor	$47250
Parts	$245.62
Total	**$47495.62**

5. Conclusion

5.1 Accomplishments

We have met almost all of our project requirements, including a portable base containing a LCD screen displaying item information, a push button turning off the notification, two tags with buzzer and Bluetooth chip communicating with portable base. When one item is out of range, the buzzer will sound immediately and the LCD will indicate the item at the same time. When the item is bringing back to the portable base, the alarm will stop beeping automatically. In addition, if the other item is out of range at this time, the buzzer will beep and the screen will display the other item information.

5.2 Uncertainties

Although we have achieved all of our requirements, we still have one uncertainty in our project. As the item is at certain distance away from the portable base, the buzzer will beep immediately. However, when the item is bringing back very fast, the buzzer will keep beeping even if it is in the safe range. The reason is that the Bluetooth chip scans the signal periodically, so that if the item distance changes too fast, the RSSI value has not updated yet, therefore the buzzer has not received the new signal to stop beeping. There are two methods to solve this little problem. One is to bring the item slowly to the portable base, so the buzzer will stop in time. Another way is to put the tag very close to the portable base and wait for a little while, the buzzer will stop itself. This is the only uncertainty part of our project, but it does not affect the function of the system because when the user brings the item back, it means that the item is already found, the goal of our system is still achieved.

5.3 Future Work

For the future work, first we would like to attach the power converter to the system to save more power for the tag side and improve the whole efficiency. If we consider to put this system for sale in the market, durable and power-saving product is certainly favorable than replacing batteries once a while.

Another possible improvement is to replace the LCD screen with touchable screen. In this way, the user can easily interact with the portable base as well as developing some other features for the system, such as entering passcode to disarm the buzzer instead of keypad, which wastes a lot of space. Also we would like to make the whole system more concrete and convenient to carry around. This needs more advanced design of PCB and more efficient power source which is very small and light, and also serves enough voltage for other modules.

Last by not least, we could improve the algorithm of our filter to make the distance threshold more accurate, or as the professor suggested during presentation, measure the FFT of the RSSI signal and find some relationship between frequency and distance, thus estimated the distance

5.4. Ethical Considerations

The purpose of this project is to build a small tracking system for personal belongings in order to avoid treasure lost. Our project will be consistent with the IEEE Code of Ethics below.

1. to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;

The small part we are going to install on the item uses coin cell battery as power supply. We will make sure that the battery makes stable performance and be safe to users. Another part on the personal item is the buzzer. We will use the smallest and safest buzzer which makes loud sound but without any possible danger.

3. to be honest and realistic in stating claims or estimates based on available data;
During the design process, we will make sure that all the calculation and verification are honest and real without any faking data.

5. to improve the understanding of technology; its appropriate application, and potential consequences;

From this project, we have learned a lot about the bluetooth and wireless signal transceiving process. With further research on different models, we have seen the improvement of technology and its application and potential consequences.

7. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

Our group will accept all constructive and useful suggestion and criticism from others towards our project in order to improve it to the best. We will also acknowledge anyone who has great contribution to this project.

9. to avoid injuring others, their property, reputation, or employment by false or malicious action;

We will make sure that the product we have developed will offer users accurate data, and will not provide any false information which may cause users injury or damage of their property.

10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

All group members will strictly follow the code of ethics during this project. We will assist each other on developing the product because we are completely responsible for our project and users safety concerns.

5.5. Acknowledgements

We would like to thank Professor Scott Carney and our TA Igor Fedorov for their valuable support and guidance during the process of our project. We would like to thank staffs from the electronic workshop Mr. Mast and Mr. Smart, as they helped producing our PCB and also providing parts for our project. In the end, we would like to thank Jeff Rowberg from Bluegiga Tech Support for his generous help regarding to bluetooth communication problems. Without all the people mentioned above, we could not achieve our project successfully.

References

Bluegiga Technologies, “BLE112,” BLE112 datasheet, Aug. 2010 [Online]. Available: http://www.glynstore.com/content/docs/bluegiga/BLE112_Datasheet.pdf

Microchip Technology. DS39951C. 2010 [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/39951C.pdf

Texas Instruments, SCBS842A. Sept. 2001, Revised Feb. 2008 [Online]. Available: http://www.ti.com/lit/ds/symlink/ri-trp-dr2b.pdf

Panasonic. Lithium Handbook. Aug. 2008 [Online]. Available: http://panasonic.com/industrial/includes/pdf/Panasonic_Lithium_CR2032_CR2330.pdf

Texas Instruments, “Mixed Signal Microcontroller,” MSP430G2x53 datasheet, Apr. 2011 [Online]. Available: http://www.ti.com/lit/ds/symlink/msp430g2553.pdf

ECE Stores, 2013 [Online]. Available: https://storesserver.ece.illinois.edu/4dcgi/catalog

Bluetooth Tutorial [Online]. Available: http://www.radioelectronics.com/info/wireless/bluetooth/bluetooth_overview.php

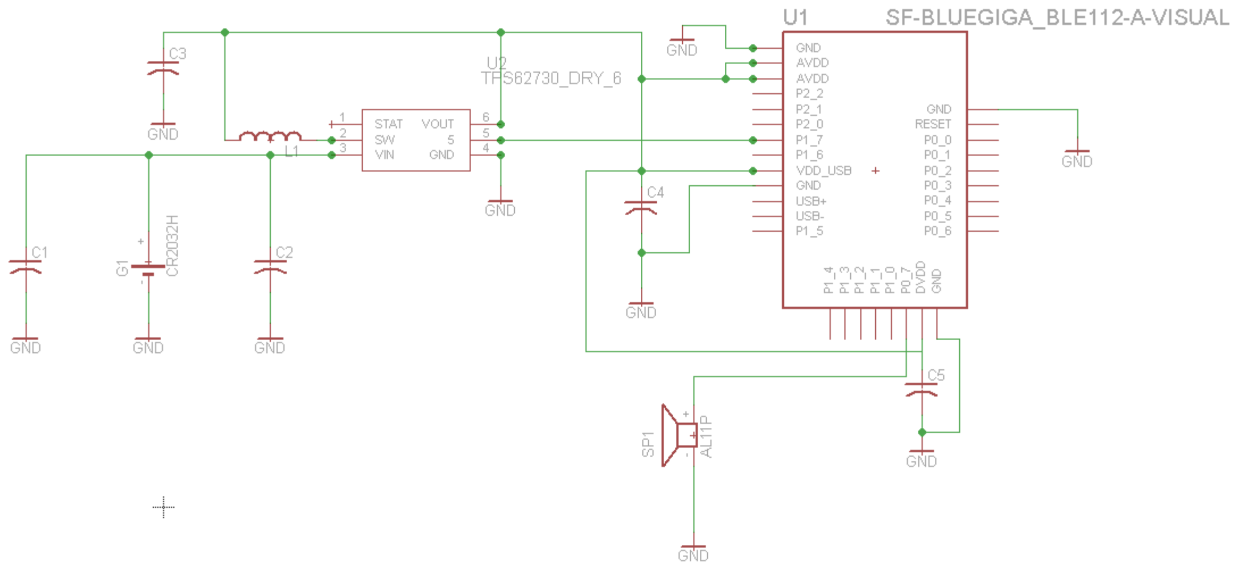
Comfile Technology, 2005 [Online]. Available: http://www.comfiletech.com/clcd216-g.aspx

CUI Inc, VBT1-SMT [Online]. Available: http://www.cui.com/Product/Resource/DigiKeyPDF/VBT1-SMT.pdf

ALL Data Sheet [Online]. Available: http://pdf1.alldatasheet.com/datasheet-pdf/view/56292/AZDISPLAYS/ACM1602.html

Appendix

Tag Schematic



Requirement and Verification Table

<u>Master-End</u>

Component	Requirements	Verification
Power Supply	1. Power Supply Module must be able to output 3V voltage to Bluetooth Chip. 2. Output 5V to Micro-controller chip and LCD screen chip. 3. Power supply operates enough power all the time.	1. Plug multi-meter on the two sides of Bluetooth Chip and check if voltage is around 3V. 2. Plug multi-meter on the two sides of battery used for microcontroller and LCD, check if it is around 5V. 3. Monitor voltage and current to make sure other Module are operating under rated voltage and rated power.
Microcontroller	1. Microcontroller is powered up correctly. 2. Microcontroller Module must be able to get information from Bluetooth Module. 3. To determine if any item is out of range for too long. 4. During an incident, it must send out the signal which could trigger the alarm.	1. (1) Check the battery is operating rated voltage 5V using multi-meter. (2) Connect the battery to microcontroller PIN4 and GND. 2. Connect the microcontroller to the oscilloscope, send signal using bluetooth, check the oscilloscope shows the signal. 3.Put one item like phone out of range (3m), connect the microcontroller to the oscilloscope and check it shows the signal from bluetooth. 4. (1) Microcontroller sends signal to bluetooth, monitor the bluetooth on the slave-end and

	5. Send signal to display the missing item on LCD screen.	check it has received the signal. (2) Send signal from bluetooth to alarm using mosfet, check the alarm sounds. 5. Connect the display to the monitor, check the signal sent from microcontroller is correct. Then check the display has showed the same information (eg. Name).
Communicatio n	1. Master-End device must be visible to each Slave-End device 2. Microcontroller must be able to send the alarm trigger signal to the Slave-End Device through the Bluetooth Module 3. Microcontroller must be able to send owner's information to the Slave-End Device through the Bluetooth Module	1. Develop test program that indicates, via UART, for the unit to be visible with given name. Pair up four slave-end bluetooth to the master end. By sending 100 data sets, test if the slave-end receive all the data sets. 2. Develop test program to verify if the bluetooth unit send out the alarm trigger signal. First, send the signal to the slave-end device only once using bluetooth, make sure the frequency is above 75dbm, check if the alarm makes sound. Then check others using same method one by one. 3. Develop test program to verify if the bluetooth unit send out correct text. First, store some text in the microcontroller like "Phone",

		then send the text to slave-end device. Connect the device to the screen, check if the text has received. At last, check if the display shows correct text.
Display	1. Display is powered correctly 2. Initialized properly 3. Display identity of the losing Item(Name) and other information (i.e password, lock/unlock) under the instruction from Micro-Controller Module.	1. Check that VDD (pin 2) is at 5V by using a multimeter. 2. Send number “1” to the display and check it displays “1” 3. Put one item (phone) out of range, power all the modules, check the display shows “phone” on the screen.
Keypad	1. Powered correctly 2. Enter the code	1. Use multimeter to test if the voltage of the keypad is 5V. 2. Enter numbers “1234” to the keyboard, connect to the display, check the display shows “1234”.

Slave-End

Component	Requirements	Verification
Power	1. Output 3V Voltage to the alarming Buzzer 2. Output 3V to the Bluetooth (slave-end)	1. Plug multi-meter on the two sides of buzzer and check if voltage is around 3V. 2. Plug multi-meter on the two sides of Bluetooth Chip and check if voltage is around 3V.

Communication	1. Slave-End Device must get the RSSI value of the Master-End device every 15 second and compare it to the reference value (-75dbm) 2. Must trigger the alarm when the RSSI value from the Master-End belows the reference value. 3. Must send signal to the Master-End to acquire owner's information 4. Must receive the text sent from the Master-End and display it 5. Must trigger the alarm when the alarm trigger signal received	1. Use the bluegiga bluetooth smart software to acquire the RSSI value and compare it to the reference value. If the RSSI value smaller than reference value (-75dbm) 2. Verify that trigger signal has sent to the alarm Method: Send signal (-50dbm) and test if the alarm sounds correctly 3. From the Master-End, verify the signal to acquire owner's information 4. Verify from the display if the received text is correct Method: First send numbers (1,2,3) to the slave end, and test if it can display, then test letters 5. Test the alarm makes sound once the signal is sent.
Alarm	1. Powered at 3V. 2. Makes sound loud	1. Plug multimeter to check the voltage is 3V. 2.(1) Power it first, stand at 1m distance, test the sound. (2) Stand 1m further every time to test the sound. (3) End test until 10 meters.

Tag Bluetooth Code:

```
dim conn
dim rssi
dim addr(6)
dim data3(1)
dim result
dim port
dim S
dim P
dim rssi_s
dim S_s
dim P_s
dim r
dim data

event system_boot(major ,minor ,patch ,build ,ll_version ,protocol_version ,hw )
         call sm_set_bondable_mode(1)
         addr(0:1) = $a9
         addr(1:1) = $12
         addr(2:1) = $2e
         addr(3:1) = $80
         addr(4:1) = $07
         addr(5:1) = $00

        call hardware_set_soft_timer(3280,0,0)
        call hardware_io_port_config_direction(0, $80)
        call hardware_io_port_write(0, $80, 0)
        call gap_set_scan_parameters($c8, $c8, 1)
        call gap_discover(2)

        P = 0
        P_s = 0
        conn = -1
end

event gap_scan_response(rssi, packet_type, sender, address_type, bond, data_len, data_data)
        if P_s = 0
                P_s = rssi
        end if
        S_s = 2*rssi+8*P_s
         P_s = S_s/10

        if P_s < 190
                call gap_connect_direct(addr(0:6),0,50,100,250,0)
        end if
end

event connection_status(connection, flags, address, address_type, conn_interval, timeout, latency,
bonding)
        conn = connection
end

event connection_disconnected(handle,result)
        conn = -1
        call gap_set_scan_parameters($c8, $c8, 1)
        call gap_discover(2)
end

event hardware_soft_timer(handle)
        call hardware_io_port_read(0, $40)(r, port, data)
        if data & $40 then
                data3(0:1) = 0
                call attclient_write_command(0,18,1,data3(0:1))
                call hardware_io_port_write(0, $80, 0)
                call connection_disconnect(conn)
        else
                if conn < 0
                        rssi = 0
                else
                        call connection_get_rssi(conn)(conn, rssi)
                        if P = 0
                                P = rssi
                        end if

                        S = 2*rssi+8*P
                        P = S/10

                        if P < 190
                                call hardware_io_port_write(0, $80, $80)
                                data3(0:1) = 1
                        call attclient_write_command(0,18,1,data3(0:1))
                    end if

                    if P > 200
                        call hardware_io_port_write(0, $80, 0)
                        data3(0:1) = 0
                        call attclient_write_command(0,18,1,data3(0:1))
                        call connection_disconnect(conn)
                    end if
                end if
            end if
end
```

Portable Base Bluetooth Code:

```
dim level
dim offset
dim value_len
dim value_data(1)
dim connected
dim result

#Initialise the display
event system_boot(major,minor,patch,build,ll_version,protocol,hw)

        connected = 0
        call gap_set_adv_parameters(32,48,7)
        call gap_set_mode(1,2)
   call sm_set_bondable_mode(1)
        call hardware_set_soft_timer($F0000, 1, 1)
        call attributes_write(xgatt_txpower,0,1,3)
        call hardware_io_port_config_direction(0, $80)
   call hardware_io_port_write(0, $80, 0)

end

event hardware_soft_timer(handle)
        if handle = 1 then
                if connected = 0 then
                        call gap_set_mode(0, 0)
                        call gap_set_adv_parameters(1600, 4000, 7)
                        call gap_set_mode(1, 2)
                        call hardware_set_soft_timer($F0000, 2, 1)
                end if
        end if

        if handle = 2 then
                if connected = 0 then
                        call gap_set_mode(0, 0)
                end if
        end if
end
```

```
event connection_status(connection ,flags ,address ,address_type ,conn_interval ,timeout ,latency,
bonding)
        connected = 1
        call hardware_set_soft_timer(0, 2, 1)
        call hardware_set_soft_timer(0, 1, 1)

end

event attributes_value(connection, reason, handle, offset, value_len, value)

        if connected > 0 then
                if handle=18 then
                        level=value(0:1)
                        if level=0 then
                                call hardware_io_port_write(0, $80, 0)
                        end if

                        if level=1 then
                                call hardware_io_port_write(0, $80, $80)
                        end if
                end if
        end if
end

event connection_disconnected(handle,result)
        call hardware_set_soft_timer(0, 3, 0)
        connected = 0
        call gap_set_adv_parameters(32,48,7)
        call gap_set_mode(1,2)
        call hardware_set_soft_timer($F0000, 1, 1)
end
```