Making A Small Range Detector System

Matthew Ramirez Lopez
Faculty of Electronics and Computer Science
University of Southampton
Southampton, UK
mrl1g21@soton.ac.uk

***Abstract*— Aiming to produce a small range detecting system, the project studies various styles of antenna and range estimation techniques. The final design ends up using a Yagi-Uda antenna which function better at mid-ranges of up to 160m.**

I. Introduction

The original intention for me with this project was to learn more about antennas and their design. To do so, I had to embed this aim into the project by making the goal of the project dependent on antennas. Thus, I decided that a suitable aim for this project was to make a detection system, similar to those commercially offered (e.g. *Apple AirTag*). However, this project's solution will focus on using antenna design to accomplish this.

II. Current Solutions

Current efforts for locating devices use differing technologies such as Bluetooth and GPS. However, they each have advantages and limitations which we will explore to inform our design.

A. Bluetooth

Usual operation of Bluetooth trackers involves notifying users of when a device is in Bluetooth range of them, and then playing a sound so the user may locate the device more accurately via hearing. More advanced techniques are now in use though, using ultra-wideband technology some trackers can provide the direction and distance of the tracker with centimetres of accuracy [1].

Predictably, the range of these trackers is limited by Bluetooth. This makes it inadequate for searching outside of its range. However, this limitation can be overcome as demonstrated by Apple. Using other phones as relays, users can now know the last known geographic location of their tracker, even if they themselves are not within Bluetooth distance [1].

B. GPS

GPS trackers have a very particular use case. Being able to sense its location by GPS its range is vast, and most importantly doesn't require the user to be physically close, unlike most Bluetooth trackers. This makes them useful for finding far away items. However, it suffers from losing accuracy when indoors. Given these characteristics, Bluetooth trackers are more adequate for locating items within a short range.

C. Decision

Attempting to replicate the types of trackers discussed to a good degree will be difficult. The products have had time to evolve and become better to the point where anything I try to do to replicate it will most likely achieve less. Exiting designs have been optimised for compactness and I do not have the components to try and make my own version at that scale.

Therefore, this project will instead attempt to use Xbee 868MHz modules with the assistance of a directional antenna to accomplish a similar functionality to existing solutions but for a different use case. The modules claim a long range line-of-sight of up to 80km (depending on antenna) and 550m in urban areas (given TX power is maxed) [2]. The application of this system could lie in finding lost devices in medium to short ranges or could be used in electronics assisted hide and seek depending on the results.

III. Design

A. Operation

For the project, two Xbee Pro S5 modules will be used which will each be connected to a Pi Pico H microcontroller. One set will act as a broadcaster, which will search for other sets by sending a message and waiting for a response (i.e. handshake). This set will be referred to as a "Seeker" and will have a custom antenna attached, this antenna's design may change as we study performance.

The other set will act as an endpoint which will be constantly listening for broadcasts to respond to, simulating a lost item (or a hider if used for playing). This set will be referred to as a "Hider" and will have an omnidirectional antenna attached.

The system in normal use would consist of one seeker and multiple hiders. The seeker would be operated by the user by pointing the set's antenna in any direction and waiting for a response, if there's a response then there should be a hider in that direction. The challenge is making the antenna as least impractical as possible for the user, whilst achieving a good performance.

B. Connection

To have the programs made for the microcontroller function with the Xbee modules, communication must first be established. In this case, Xbee modules use UART connections. Furthermore, it requires a supply of 3.3V to function. To satisfy these requirements, specific pins on the Pico must be bridged to the Xbee as shown on Fig. 1. Xbees have multiple modes, but I will opt to use AT mode (a.k.a. transparent mode) for simplicity.

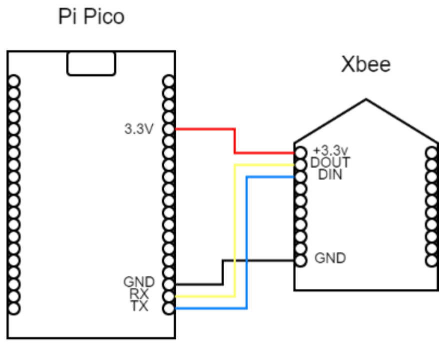

Fig. 1. Connections between Pi Pico H and Xbee Pro S5

Before connection however, the Xbee modules must be configured. Baud rate, parity, and stop bits must be known prior to connecting the components, so they must be set by me. To do so the Xbee modules are connected to a computer and are configured with XCTU, a tool provided by *Digi*. Not only will UART connection parameters need to be set here, but also the Network ID and addresses that the modules are allowed to communicate with. The Network ID will have to be the same across all modules in order for them to communicate between each other. The addresses are set to all, in other words, to broadcast. This will allow for modules to be added into the system easily.

IV. Potential Antennas

The antenna to be used by the seeker module is dependent on the antenna's practicality and performance. The main factor that will be affecting practicality is the antenna's size. Performance will be determined by the antenna's gain and directionality.

A. *Yagi-Uda Antenna*

Consisting of a driven element, which can be a dipole, and additional parasitic elements (including a reflector) the Yagi antenna has a simple structure which can be easily reproduced [3]. Its key advantage is the ease of improving performance by only adding elements. However, the elements' required distance from each other can lead to a large apparatus, this is compounded further by the dipole's length being inversely proportional to frequency. The parameters such as the number of elements and the choice of using a can reflector will be explored in the project to find an optimal balance.

B. *Patch Antenna*

Having the capacity to be made small, this antenna has great potential in situations where space is a strained. Their limited individual performance can be remedied by grouping them up to make microstrip antennas, which have the further potential of being made into an array antenna with higher gain. There is size dictated by the properties of their dielectric material, specifically the dielectric constant.

C. *Helical Antenna*

Normally being omnidirectional (typically called "Rubber Ducks"), this antenna can be modified to be directional. The antenna can use one or more wires wound in the form of a helix to receive and transmit waves. To be made directional it requires its circumference to be close to the that of the wavelength and it requires a ground plate so only one end transmits waves. The size increase due to this would be significant.

D. *Parabolic Antenna*

Most prominently used for TV, the parabolic reflector antenna simply reflects all waves with a parabolic shaped reflector to a single focal point where a dipole can be placed. Reproducing it is more difficult due to the accuracy needed when making the reflector. The directionality of the antenna is also proportional to the size of the reflector, making improvements more costly because a new dish would have to be made.

E. *Can Antenna*

The "Cantenna" is an extremely simple antenna, consisting of a metal can (or metal coated material) with the element inside. For the lower frequencies the can's diameter will have to be increased. This design can be combined with other antennas such as the Yagi-Uda to increase directionality, a possibility which will be considered.

F. *Corner Reflector Antenna*

Similar in concept to the parabolic antenna, the antenna has two reflectors (can be solid sheets or meshed) at 90° to each other. The driven element is placed at a distance between the reflectors. The design's components' size is proportional to the operating wavelength.

V. Making the Antenna

Various types of directional antennas have been identified but not all are adequate for use in the project. Antennas which will be considered are those which are simple to recreate and/or are not excessively impractical (i.e. too large). Antennas which conform to those conditions are the Yagi-Uda and can antennas.

A. *Operating Parameters*

As all the dimensions of the antennas are dependent on the operating wavelength, it must be calculated. This can easily be done by using wavelength formula (1).

$$\lambda = v/f \tag{1}$$

Where v is the wave speed and f is the frequency. As wave speed is constant (at approximately 3e8m/s) we simply substitute *f* for 868MHz, equating to 345mm.

B. *Yagi-Uda*

For the antenna it was decided to only use 4 parasitic director elements (excluding the dipole and reflector) to keep the size of the antenna reasonable, this is also due to the benefits of adding more directors diminishing after 3 or 4 [3]. The dimensions of the antenna are mainly gathered from a helpful online tool and are shown in Fig. 2.[4]. The tool has the directors' length and spacing vary, which a study suggests provides better radiating performance [5]. The boom will be made of wood to insulate the parasitic elements from the driven element, the dipole of choice for this antenna will be a straight type.

C. *Can Antenna*

After making a mock version of the antenna with a simple pringles can I didn't find very satisfying results. The diameter of the can needed to be greater (at around 20cm) to see a better result. However, I had difficulty finding something of that size so I had to put this antenna design off.

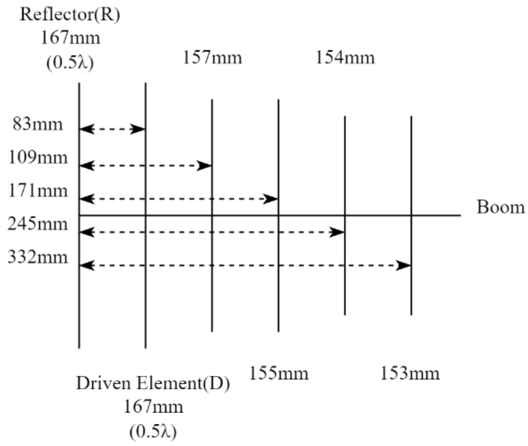

Fig. 2. Yagi-Uda Antenna Dimensions Diagram

VI. Range

A. System Range

The effective Range in urban environments is limited but in open environments the tool has greater capabilities. Estimating an effective range for either environment proved to be difficult, the first attempt used the Friis transmission equation rearranged for range (2). The gain values were chosen to be 2dB for the receiver as that is what the rubber ducky antenna claims and 9dB for the transmitter as that is what the Yagi-Uda is estimated to provide. The minimum power that must be received is -112dBm.

$$R = \frac{\sqrt{\frac{P_T G_T G_R \lambda^2}{P_R}}}{4\pi} \tag{2}$$

This model gives an excessive result of 388km. The large value is most likely because of the omission of loss factors such as atmospheric interference, multipath etc. For a better estimation, these factors must be accounted for, even if they are only rough estimates. Thus, the link budget equation is repurposed (3).

$$R = \frac{10^{\frac{(P_{TX}+G_{TX}-L_M+G_{RX}-P_{RX})}{20}} * \lambda}{4\pi} \tag{3}$$

However, finding an appropriate L_M (losses) to simulate an indoor or urban area was difficult. Unlike for higher frequencies, there is no empirical data from studies on the propagation of 868MHz RF to model after. Common models such as the Okumura and Hata models are unapplicable to the project's operating frequency. Due to this I attempted to find the loss Digi used in their promotional material. As they claimed 550m indoor/urban areas at what I am assuming are optimal conditions for the antenna (i.e. fully powered), I calculated that the loss factor value they used as 55db. Using this value, the new estimate for indoors is calculated to be 308m due to the transmission power being reduced (because of the Pico). However, the effect the antenna I will be using will have on this is yet to be known.

B. Detection Range

Notifying the user of the distance between them and the detected module will aid them. A common approach which that will also be followed here is that of using the Received Signal Strength Indicator (RSSI). By measuring the received signal's strength, we can use the inverse-square law to estimate distance. This estimation is susceptible to many other factors, but due to the antennas directionality I hope the effects are less drastic. However, if returned values are too inaccurate, then I will opt to use values to show proximity instead of an actual distance which may end up being wrong. So, values will be grouped to represent proximities like "Close" or "Far" instead.

The formula that will be used to estimate distance will require the RSSI at 1m as a control variable and an environmental factor to use as a loss factor, which ranges from 2 to 4 [6]. There are many variations of using RSSI for distance estimation, but I will stick to the simple version described in [6] (4).

$$D = 10^{\frac{M-RSSI}{10*N}} \tag{4}$$

Where M is the measured RSSI at 1m and N is the environmental factor.

VII. Implementation

A. Code

The implementation of the handshake system that was proposed will depend on what metrics will be measured. For a simple game of hide and seek, range estimation is most likely not going to be necessary. Therefore, a simple implementation will be made. It will simply notify the seeker if they are pointing in the right direction via the LED on the microcontroller.

```
def loop:
    while 1:
        time.sleep(0.5)
        led_pin.off()
        uart_xbee.write('marco')


def waitForResponse():
    time.sleep(0.2)
    response = uart_xbee.read()
    if response = b'polo':
        led_pin.on()

loop()
```

```
def loop():
    while 1:
        time.sleep(0.1)
        data = uart_xbee.read()
        if data == b'marco':
            respond()

response = 'polo'

def respond():
    uart_xbee.write(response)

loop()
```

Fig. 3. Simple handshake system, Seeker and hider code respectively.

For finding lost items however, the simple implementation shown in Fig. 3 is not enough. A range estimation will be necessary in this use case and so RSSI will have to be taken from each response. This will involve communicating with the Xbee module in command mode as shown in Fig. 4.

An issue I suspect will happen is that users will have difficulty determining the direction of a response signal when they are close to the hider. This will probably be even worse given the lower frequency of operation, which gives it greater penetration, and therefore range. I will most likely find the distance at which this effect starts to occur during testing.

```python
def loop():
    while True:
        time.sleep(0.5)
        if uart_xbee.any():
            data = uart_xbee.read()
            print("Something was received!")
            print(data)
            if data == b'marco':
                led_pin.on()
                respond()
                rssiCalc()
                led_pin.off()


M = -60#TO BE MEASURED #RSSI at 1m
N = 2 #Environmetn Loss Factor
def rssiCalc():
    time.sleep(1.1)
    uart_xbee.write(b'+++')
    time.sleep(1.1)
    uart_xbee.read()#flushes 'OK' response from Xbee entering command mode
    uart_xbee.write(b'ATDB\r')
    time.sleep(0.1)
    if uart_xbee.any():
        rssiVal = uart_xbee.read()
        numVal = int(rssiVal.decode().replace("\r", ""), 16)
        estimatedDis = 10 ** ((M + numVal)/(10*N))
        print(estimatedDis)
    uart_xbee.write(b'ATCN\r')#exit command mode
    time.sleep(0.1)
    uart_xbee.read()


response = 'polo'
def respond():
    uart_xbee.write(response)
    print("Response sent...")

loop()
```

Fig. 4. RSSI tracking implementation, Seeker code.

B. *Build*

With the Yagi-Uda antenna built I put all the components together as shown in Fig.5. However, I only had a single battery. This meant one module would have to connected to my computer and be static during testing. This was fine since that is what I intended anyway. Something I would have liked is a small display to show the estimated range but for now I will simply print it to the computer.

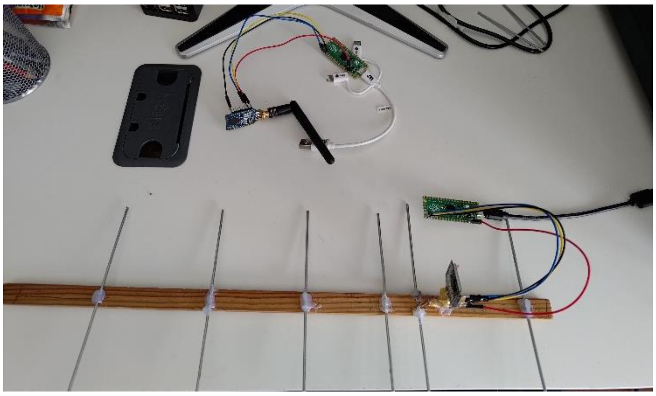

Fig. 5. Seeker (Bottom) and Hider (Top) modules assembled.

VIII. Testing

To measure the performance of the antenna, the hider will be placed in an indoor location in an urban environment. The seeker will then be measuring RSSI and estimate distance at increasing intervals. Eventually responses will cease to be received, singling the maximum range of the system has been reached.

In addition to range, directionality will also be measured at each range. This will be done by recording the angle ranges at which the antennas pick up signals.

A. *Results*

As predicted before, the seeker would detect the hider no matter what direction it was pointing at once in close proximity. The issue was that it occurred everywhere in the building. However, once I removed the antenna from the hider to reduce its gain, I capped this effect at around 7m. Doing this also limited the indoor range though, at around 20m.

Due to the issue, once I began tests with both antennas attached, the results for the first couple of distances were not that great as seen in Table 1. The Xbee modules seem to not have very accurate RSSI readings, the values it outputs all seem to be almost random. The directionality seemed to improve with distance which was contrary to what I thought was going to happen. In an open environment perhaps performance and estimation could work a lot better. Getting a response after 160m was extremely difficult so I will mark that as the limit of this system.

TABLE I. Antenna Distance Estimation

Metrics	Actual Distances(m)					
	1	***5***	***10***	***20***	***30***	***40***
Estimated Distance (m)	1	1	1	11	22	19
Angle Range (Degrees)	360	360	360	360	360	360
	50	***60***	***70***	***100***	***125***	***150***
Estimated Distance (m)	31	56	50	100	35	50
Angle Range (Degrees)	270	180	90	5	5	5

IX. Conclusion

From test results the end product seems to be fairly bad at range estimating and directing at close range. The low operating frequency of the modules makes the signal have a longer range which makes finding the direction of it only viable after a certain distance. The range estimation was extremely inconsistent so the feature will have to be abstracted to only give vague proximities as the values were too inaccurate to be informative. To remedy this, I could try and use a different antenna for the seeker module next time.

Although not excessive, the sizes of the seeker antenna is significant enough to make the tool impractical for use as items trackers in usual cases. However, a possible use case for the tool may lie in wireless sensor networks. Modules in WSNs may end up missing due to weather or other causes and if they do not already have GPS trackers, recovery can be possible with the tool. With the right configuration this could work.

Potential improvements for the system are the antenna reflector, possibly using a can in its place to create a hybrid

Yagi-Can antenna. I suspect that this would have a lot better directionality than what I have achieved with the 6 element Yagi. More adjustment of the RSSI distance estimation parameters can be done to get better predictions too.

References

[1] Apple, “AirTag,” *Apple (United Kingdom)*, 2021. https://www.apple.com/uk/airtag/ (accessed May 01, 2024).

[2] “User Guide XBee ® /XBee-PRO ® 868 RF Module.” Accessed: Apr. 18,2024.[Online].Available:https://www.digi.com/resources/documentation/digidocs/pdfs/90001020.pdf

[3] W. L. Stutzman and G. A. Thiele, *Antenna theory and design*. Hoboken, Nj: Wiley, 2013.

[4] A. C. F. Changpuak aka, “Yagi Uda Antenna (Rothammel),” *www.changpuak.ch*. https://www.changpuak.ch/electronics/yagi_uda_antenna_DL6WU.php (accessed Apr. 15, 2024).

[5] A. Raji, “IMPACT OF ELEMENT LENGTHS AND SPACINGS ON THE PERFORMANCE CHARACTERISTICS OF YAGI-UDA ANTENNA”, *laujet*, vol. 14, no. 2, pp. 25-35, Nov. 2020.

[6] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, “A Location Estimation Algorithm Based on RSSI Vector Similarity Degree,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 371350, Jan. 2014, doi: https://doi.org/10.1155/2014/371350.

Appendix

Appendix A

The following is the code used to establish an UART connection between the Picos and Xbee modules.

```
import machine
import time

uart_xbee = machine.UART(0,baudrate=9600,tx=machine.Pin(16),rx=machine.Pin(17))
uart_xbee.init(bits=8, parity=None, stop=1)
led_pin = machine.Pin(25, machine.Pin.OUT)
```