Lab 08: Real-Time Wireless Navigation

When You Finally Lose The Tether

INTRODUCTION

On December 12, 1901, Guglielmo Marconi was the first person to send a transatlantic wireless (radio) signal. His test message, in Morse code, was just the letter 's' (dot dot dot). This was simply proof-of-concept: the transmitter did what it was supposed to do, and the receiver did what it was designed to do. And the rest, as they say, is history.

Try to imagine the inconvenience of wires…wait, you don't have to , do you? Up until now, it's been a pain in the neck, but the bots had to be tethered to the computer, either to load a new program, or continuously if you wanted to drive in real time using keystroke commands. If you've ever played with an RC car, you already know there's a better way.

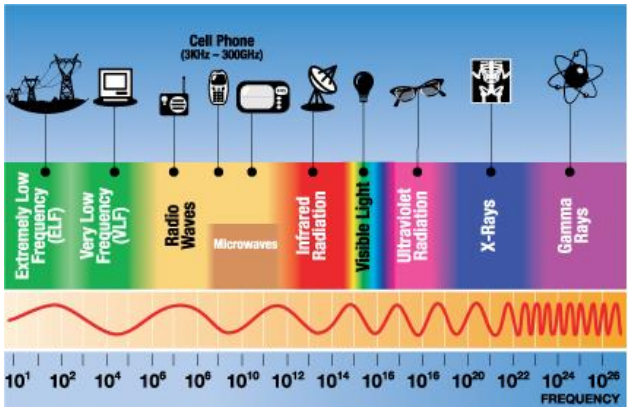
The simplest way to control your BOE-Bot is still with keyboard commands. What we need is a way to send those keystrokes to our BOE-bot wirelessly. The XBee is a simple wireless RF device capable of just that: sending and receiving signals. Conveniently, you have a pair of them–one for each BOE-Bot.

OBJECTIVES

The specific goals for this exercise are:

- Modify an existing algorithm to complete a similar task
- Learn more advanced branching and looping structures
- Assemble, install, and test the XBee RF devices
- Program your bots to communicate wirelessly
- Compete as teams in a soccer-style game

THE BETTER WAY!

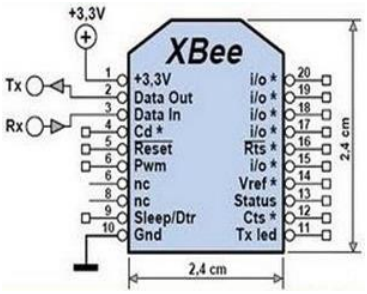


You also know, if you've used anything from a TV remote to an RC car, that it's a two-way proposition: transmission (something sends a signal) and response (something else receives the signal and can do something with it).

Notice that all of the 'wireless' devices you are familiar with use the same basic idea, but at a different frequency. An RC car, for example, operates in a range of frequencies between 27 and 49 MHz. A TV remote operates at 38 kHz, and a garage door opener in the range of 300-400 MHz. A 3G phone operates in the 800-850 MHz frequency band, and your 5G could be anywhere from 2.5 to 3.7 GHz, depending on your carrier!

Because the spectrum is continuous, and the frequency range is enormous, it can support a huge number of devices—as long as everyone knows which frequencies are allocated for which uses. The XBee transmitter/receiver pairs operate at 2.4 GHz.

WIRE IT UP!



Easier than it looks! The chip has a lot of pins, but we only need four: power (1), transmit (2), receive (3), and ground (10). We will wire the XBee directly to the 20-pin female header on the BOE board.

Connect the first wire from XBee pin 1 (+3.3V) to the header receptacle which plugs in to VDD. Next, XBee pin 2 (Tx = black wire) to receptacle which plugs in to P0 and XBee pin 3 (Rx = yellow wire) to receptacle which plugs in to P2. Lastly, the XBee pin 10 (Gnd) should be wired to receptacle which plugs in to VSS. That's it. You're ready to go wireless!

CONSTRUCT THE ALGORITHM

We already have a bot that we can control with keystrokes; those keystrokes, however, are delivered directly to the bot via wire. The basic algorithm for controlling the bot does not change, but the method of delivering the signal does. We want to lose the wire and have a bot which roams freely.

Notice that you have an XBee for each of your BOE-Bots. What we will do, then, is keep one bot (HomeBot) tethered to the computer. That bot will accept the keyboard input, then transmit it wirelessly. Conveniently, the second bot (RoamBot, also wearing an XBee, but not tethered by a USB cable) will receive the signal and execute the command.

This means you will need two programs: a Transmit program to accept the user input and broadcast it, and a Receive program to accept the RF signal and interpret/execute the command.

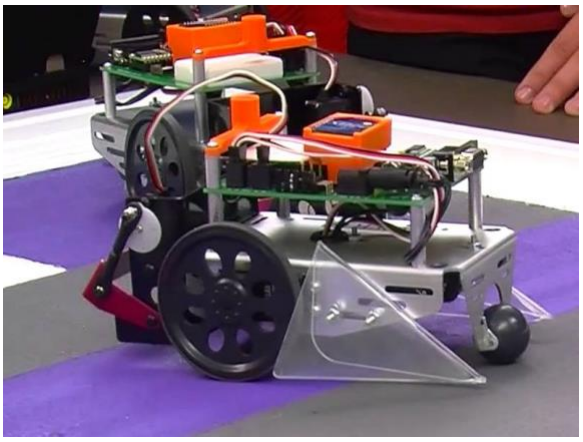
WRITE THE CODE

1. Download the code. The file transmit.bs2 contains the source code you need to get started. The file is in the ENGR 1301 Google drive, or can be downloaded directly from the course web. This should compile and run cleanly and should ultimately be loaded onto and run by the HomeBot. The only modification required will be to set your comm channel to a unique value.
2. Install the XBees. Double-check that your XBees are properly wired. Triple check that you do not have the power and the ground reversed!

3. **Add to the existing.** Your Receive program should be based on the keystroke control program you have already completed. Make a copy and rename to preserve your original program, then modify the copy to accept the RF input.
4. **Check your subroutines.** These should already be written and debugged, but if you have not done so already, you should check that they operate as intended. These will be the basis of your Receive code (remember, RoamBot has to know what to do with the input signal).
5. **Follow the pattern.** Examine the Transmit code and think about what must be added to your keystroke control code to make it a Receive program. For example, since both bots wear XBees, they both need the same PIN declarations. Should they also be on the same channel? Communicating at the same rate? If the transmitter sends the signal using a SEROUT (serial output) command, what do you suppose the syntax is for receiving it as serial input?
6. **Compile and test.** When you have completed the Receive code, make sure it compiles cleanly. In order to test the wireless communication, both bots must have the appropriate code loaded and ready to run (I/O switch in position 2). Which bot has to be code-loaded first (Hint: Not the one that has to stay tethered to the computer!)

PROGRAM THE HIPPO JAW

Again, we do not have enough attachments to give every student a hippo jaw. Extra robots are provided with jaws and a wireless XBee already attached.



7. **Create a new program.** You should now have your own wireless receiver program. Do not overwrite or modify it! Use a copy of this program and alter it as necessary.
8. **Which way is forward?** When you examine the putter, you should notice that your bot is back in front-wheel drive mode.
9. **Control the Jaw.** Raising and lowering the jaw is similar to the putter, with one exception. The hippo jaw servo is not the same as the servos which drive the wheels. The PULSOUT pin,spin is still the proper command, but the spin arguments need to be tweaked because the servo is not a continuous rotation model. The servo can only turn through 180°, not a full 360°.
10. **Demonstrate your success.** Once you have a working jaw, use the football pitch to show the bot in action. Use your phone to document with a short video clip.

SAVE AND SUBMIT

Be sure to save your source code frequently! Always save to your own UCA Google drive. If you are using the UCA computers, save the program on the Desktop—but always save it in a second location. If it's on your Google drive, you won't need to be in the lab to access the files!

- **Demonstrate your success:** Practice navigating and putting, then use your phone to document your skills with a short video. You should be sure that your video shows clearly who is controlling the Bot, and that the Bot operates properly (forward, backward, left and right, grabbing with the jaw). It does not need to be long or fancy, the video simply needs to demonstrate that you have achieved the objectives of the exercise. Blackboard Assignments supports the submission of multiple video file formats, so you should be able to upload your video with your source code.
- **Use the proper file name:** Whatever you have named your program in the source code, you must use the correct filename for submission. Name your receiver program lastnameLAB08, obviously using your own last name. It should already/automatically have the .bs2 file extension. You do not need to submit the Transmit code that was provided for you.
- **Submit electronically:** All programs are due no later than 6:00 PM on Tuesday, 26 March 2024. You must submit via Blackboard.

GRADING RUBRIC

Your performance will be evaluated using the rubric below:

ASSESSMENT	CRITERIA / VALUE		POINTS EARNED
Lab 08 Program: GrabAndGo! DUE: Tue 26 Mar 24 (filename: lastnameLab08.bs2)	Compilation: Program compiles cleanly	5 points	
	Annotation: Program is sufficiently commented	5 points	
	Execution: Correct loop structure, bot drives wirelessly	7 points	
	Jaw: Hippo jaw operates properly	6 points	
	Success: Operation is completely documented via video	7 points	