Arduino RF/Wireless Workshop
Hosted by Bloominglabs and The Collaboration Room

Course Outline:

I. Getting started - verify gear is working and get students familiar with the hardware and software

A. Each pair of participants should have a unique station number, 1-n (to be used in section III).
B. Connect USB and/or power to your Arduino
C. Load the blinky sketch up in the Arduino application and upload to your Arduino board
D. Verify this process is working, you should have a blinking LED now
    Troubleshooting:
    1. Did the Arduino application give an error?
        a. verify the correct board type is selected
        b. verify the correct serial port is selected
        c. reset the arduino and try again
        d. unplug the arduino, then plug back in. restart the Arduino application. try it now.

    2. The application said the sketch uploaded, but nothing is happening
        a. verify the led is wired up correctly and check polarity

E. Instructor: make sure every group has two blinking LED's before moving forwards

II. Lets do some RF!

A. Wire up the TX and RX boards, one per arduino (remove power from board first)
B. Double check that power is wired correctly
C. Type in or copy/paste RF counter code into Arduino application (TX for transmitters, RX for receivers)
D. Power up boards, upload code
    1. REMINDER!! RX boards MUST disconnect data-line to Arduino before code can be uploaded, otherwise strange errors will result

E. Participants with RX boards, activate serial port monitor
    1. Do you see numbers counting from 0 to 255? If not begin troubleshooting:
        a. make sure boards are side-by-side so they are in RF proximity to each other
        b. verify participant with TX board is using TX code and participant with RX board is running RX code
        c. verify wiring of power lines and data line to TX/RX board
        d. verify baud rate
        e. ask instructor for the scanner and how to use it
    f. ask other groups to stop transmitting briefly to ensure there's no interference

F. Instructor: make sure every group is now sending and receiving data before moving forwards

III. Experimentation – Instructor led

IV. Sending and receiving sensor data

A. Wire up 5k potentiometer on the TX breadboard to Arduino pin 3
    1. Add-in code to read input analog data and provide a number from 0-9
    2. Verify RX board receives data
    3. Leave potentiometer wired up

B. Wire up light sensor on TX breadboard to Arduino pin 5
    1. Add-in code to read light sensor data and send out TX
    2. Verify RX board receives data

V. RF Distance Testing

A. Groups should finalize the code they want to run on the TX board for this test
    1. Load code on TX Arduino
    2. Disconnect USB from this Arduino
    3. RX Arduino stays connected to computer with USB - Arduino application should be in serial port monitor mode

B. Carefully wire up 9V directly to TX boards rather than 5V from Arduino regulator as we have been using
C. Add antenna wire to TX and RX antenna pins (315MHz gets 9.5” wire, 434MHz gets 7” wire)
D. Connect 9V battery to TX Arduino
E. Start testing distance, TX Arduino is now free to be moved around the room
    1. other groups should disable their TX boards while final distance testing is being performed :)

VI. Cleanup

A. Backup and save any code from the computer that you wish to keep
B. Disconnect all components from the Arduino breadboards, put back into bags
C. Put Arduinos back into containers, account for all pieces
D. Fill out questionnaire
E. Have a great night, thanks for coming!!

Appendix I. Going Further on your own:

1. Connect a motion sensor, have a basic motion sensor alarm for your house or the fridge
2. Make the transmitter control an application on your computer
3. Control a circuit on the receiver from your computer
4. Make a basic Arduino 2-player game
5. More extensive RFDriver library - VirtualWire: http://www.open.com.au/mikem/arduino/

III. Experimentation: this section to not be handed out to students until after completion, instructor should lead each step

A. Students with TX boards should increase delay at end of their sketchs from 10ms to 20ms
1. What happens on the receiving end (serial port monitor)?
2. Turn the delay back down to 10ms, verify receiver works again
3. Now turn up the delay to 50ms, what happens on the receiver end?

4. Instructor: if available, turn on the oscilloscope and connect RX receiver to demonstrate what is going on
a. start by showing receiver output with no TX, show the RF background noise
b. have student turn on a compatible TX board with 10ms timing, show the RX data output
c. have student turn up delay to 20ms on TX board
1. noise should be visible between TX packets, if not increase to 50ms or higher until this is seen
2. the "gaps" in transmitted packets are where the garbage data is coming from on the RX side
3. when the delay between packets is 10ms then there are no gaps for RF noise to creep in

B. Students should work together to make changes to their code to reduce errors
1. Suggest some schemes that could work
a. add header to tx code and detection code to rx code
b. add rx code to eliminate all but valid range of input
c. students may not have C experience, may need to have some easy sample code on-hand to use
d. code to send alpha would be good to

2. Students should have about 20-30 minutes to experiment at this point
3. Watch for groups that are struggling or off-course. Point them towards some example code if they need help.

C. After 20-30 minute period has passed, introduce students to the RFDriver library
1. First, what solutions did each group come up with? Did it work? Well?
2. Obviously someone else has already dealt with these problems before, how did they overcome them?
3. What does the RFDriver library do for you?
4. Why does it make your life easier, why is it more reliable than the code you've used so far?
a. adds initialization header (pre-amble)
b. rudimentary network address
c. CRC check
d. all you do is read or write to a variable, the rest is handled for you

5. Pass out RFDriver.pde on USB sticks or over the network (or hidden on each system)

6. Have students reload sketches BHRF001-II-TX/RX onto their appropriate boards
a. drop RFDriver.pde into their sketch directories for BHRF001-II-TX/RX, save and reload sketch
b. have students manually modify the BHRF001-II-TX/RX code so it's using the RFDriver library

c. have each pair modify their Network Identifier to be their group number
d. verify each pair of students is sending and receiving data
e. have students turn up TX delay to 500ms and verify they are cleanly receiving data
f. now have students move their boards around so TX and RX boards on the same frequency are interfering with each other
1. they should notice that opposite groups are not receiving data from each other
2. it's possible that they could still be interfering with each other, may need to increase TX delay ms
3. ask students if they understand what is happening - network ID of the RFDriver code

D. Instructor: do not move on until all students have gotten through the last steps.