

day77 micro:bit radio
Due: Wednesday 4/3/24

Today we're going to use the micro:bit built-in Bluetooth radio. You may be familiar with Bluetooth radio which can connect our phones to wireless earbuds or speakers, or allow a phone to talk with a car, or even a phone to talk to another phone when you are nearby. Each micro:bit has Bluetooth on it, and we can access the functionality with a few simple commands:

```
from microbit import *
import radio
radio.on()
while True:
    incoming = str(radio.receive())
    if incoming != "None":
        display.scroll(incoming)
```

I put this program in a starter file attached to the assignment. Please open it, copy it, and put it in a new program, then save it with your name in a comment at the top.

The new commands are the `import radio` line, the `radio.on()` line, and the `radio.receive()` function. Look at the above code and see if it makes sense to you. I explain it in the next paragraph, but you should try to see what you think it is doing first.

Explanation: We start the radio, then go into an infinite loop checking if anyone has broadcast anything (`radio.receive()`). If nothing comes in, Python gives us "None". So we check if it's not "None" and then scroll what we received across the display if it's not "None".

We will use one other command today: `radio.send()`, which you could use as shown below in the last two lines added to the above program:

```
from microbit import *
import radio
radio.on()
while True:
    incoming = str(radio.receive())
    if incoming!="None":
        display.scroll(incoming)
    if button_a.was_pressed():
        radio.send("Hays")
```

The only change from the original program is that if I press `button_a` I send "Hays" out on the radio. Anyone else running this program would see "Hays" scroll by on their display every time I press the A button. Add those two lines, but have it send your first name. Flash, and try it out. See if you can find someone in the class who sees your name scroll by.

The challenge here today is that all of you in the class will be sending and receiving on the same radio frequency, so we're going to divide into subgroups. (If you use the block based programming interface for the micro:bit you can set your micro:bit to only send and receive on a specific channel. The Python coding interface we have does not allow that, so we'll be coordinating in a different way.) I'm breaking you up into small groups.

See the top of the next page for details.

Groups:

Zane, AJ, Janell, Rob
Roy, Matthew, Mitch, Kailer
Oscar, Theo, Mathew
Thunder, Kaz, Kevin
Emma, Milla, Avery
Noah, Alejandro, Gil, Simran

Today's task:

Write a program that loops forever and inside the loop does the following:

- When you press the A button your micro:bit sends your name, exactly as written above in the group listing.
- Your micro:bit also listens for incoming radio messages. When something comes in check if the incoming text is the name of someone in your group. If it is, report that (you can just `display.scroll()` their name). If it is not the name of someone in your group do nothing with it.
- When you have heard from everyone in your group at least once, show a smiley face for 5 seconds, then start the whole process over again.
- Note: each person only needs to worry about all of the OTHER people in their group; obviously your micro:bit doesn't need to wait to hear from you.
- **Do NOT scroll what you received over the radio if it is not someone in your group.** This is to avoid everyone's messages from crowding out each other on everyone's micro:bits.
- Help your group mates if you get your code working before they do. Work together, you're a group after all and you need everyone to be functioning to get checked off.

For 8/10, do the above.

For 10/10, make your micro:bit do a unique animation of some kind while it is waiting to hear from people in your group. Each person's animation must be different. Can you do something using `i.set_pixel(c,r,9)` and a for loop? Remember, the following code gives you the numbers 0 through 4 in the variable `r`...:

```
for r in range(5):
```

You can do any kind of animation (for reference, you can go see what you did on the dice program, but please do something different here today.)