

day94 micro:bit radio

Due Thursday 5/5/22 (6th period), Friday 5/6/22 (7th period)

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)
```

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.

Explanation: We start the radio, then go into an infinite loop checking if anyone has broadcast anything (`radio.receive()`). If nothing came 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("hello")
```

The only change from the original program is that if you press button A we send "hello" out on the radio. Anyone else running this program would see "hello" scroll by on their display every time you press the A button.

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 have to divide us all up 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:

6th period:

- 1: Oren, Traver, Pono, Austin
- 2: Dylan, Sam, Niko, Daimen
- 3: Isaac, Adan, Preston, AJ
- 4: Larry, MichaelF, Eli, Wyatt
- 5: Xavier, Ethan, MichaelL
- 6: Evan, Rowan, Haven
- 7: Quincy, Esme, Miguel, Antonio

7th period:

- 1: McKenna, Memo, Raegan
- 2: OGHenry, Alexis, Eli, Colin
- 3: Miguel, Landon, Ben, Randy
- 4: Ransom, Bryce, David
- 5: Josh, Matthew, Aidan
- 6: Henry, Bardeaux, Caden

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 (6th period: Michaels, take note, and 7th period: Henrys, take note).
- 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.
- Also note: don't 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 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):
```