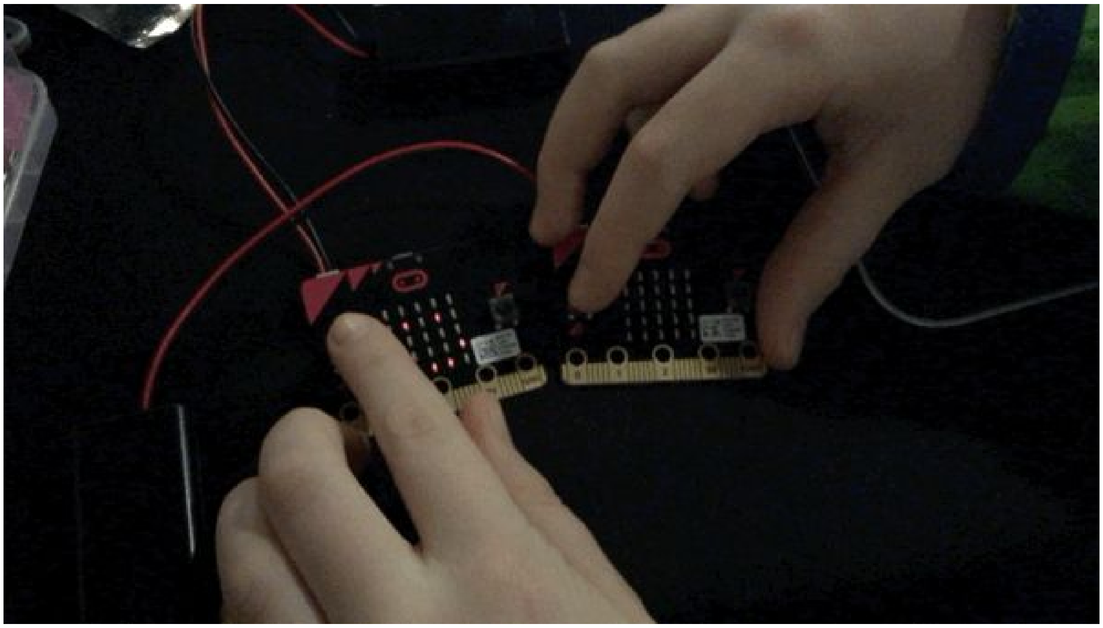
Micro:bit Firefly Messenger



Learn how to send messages to your friends on the other side of the room. Or broadcast an animation or emoji to a group of people at the same time. This guide will show you how to pass messages between micro:bits using Bluetooth and the micro:bit's radio.

Get started:

- Connect your micro:bit to your computer using the USB cable
- Download and install the Mu Editor from https://codewith.mu. You'll use this to write the code for this project.

Get coding:

Here is the Python script for the Firefly Messenger. You can type it out or copy-and-paste it into the Mu Editor. Then to load it onto your micro:bit press the "Flash" button:

```
#---------START----------------------------------
# A micro:bit Firefly.
# By Nicholas H.Tollervey. Released to the public domain.
import radio
import random
from microbit import display, Image, button_a, sleep

# Create the "flash" animation frames. Can you work out how it's done?
flash_img = [Image().invert()*(i/9) for i in range(9, -1, -1)]

# The radio won't work unless it's switched on.
radio.on()

# Event loop.
while True:
        # Button A sends a "flash" message.
        if button_a.was_pressed():
        radio.send('flash')  # a-ha

        # Read any incoming messages.
        incoming = radio.receive()

        if incoming == 'flash':
        # If there's an incoming "flash" message display
        # the firefly flash animation after a random short
        # pause.
        sleep(random.randint(50, 350))
        display.show(flash_img, delay=100, wait=False)

                # Randomly re-broadcast the flash message after a
        # slight delay.
        if random.randint(0, 9) == 0:
                sleep(500)
            radio.send('flash')  # a-ha

#-----------END-----------------------------------
```

Once you and at least one other coder at your desk have loaded the code onto your micro:bits you can try it out :
Press button A to send out flash-messages and see what happens!

Remix the Code:

You can change the code to add more features or make it more personal to you. Try out these suggestions.

Personal Avatar
Bored of the look of the firefly image? Create your own image. It could be an emoji, your initials, or a symbol. This tutorial has an explanation of how to make images and examples of the micro:bit’s built-in images:
http://microbit-micropython.readthedocs.io/en/latest/tutorials/images.html

To make a new image you'll need to change this line of code (line number 8):

```
flash_img = [Image().invert()*(i/9) for i in range(9, -1, -1)]
```

Make your own channel

Look at the passage in the Radio tutorial where the author talks about "Addressing":

"The problem with radio is that you can't transmit directly to one person. Anyone with an appropriate aerial can receive the messages you transmit. As a result it's important to be able to differentiate who should be receiving broadcasts."

This means your micro:bit receives any message sent from nearby micro:bits. Each time it gets a message it stores it in a variable called `incoming` (see line number 19)

```
incoming = radio.receive()
```

But your Python script ignores any messages that aren't equal to the string 'flash' (see line number 20)

```
if incoming == 'flash':
```

You can use this feature to set up your micro:bit to only display messages from your friends. Agree on a secret word to identify messages from the group, and change the code to use this instead of 'flash'. Everyone in your secret group will need to do this and load the new code onto their micro:bit. Hint: you'll need to make changes to line 17 and line 20)

Walkie-Talkie

Create a walkie-talkie channel to exchange text messages with a friend instead of just triggering an animation or picture. Have each message start with your secret word so that you can identify which messages to show.

Here are two bits of code that could help you:

The `in`-operator:

```
if "blah" in somestring:
 print (somestring)
```

Learn more about it here: http://pythoncentral.io/quick-tip-using-pythons-in-operator/

The slice-operator:

```
newString = somestring [5:]
```

Learn more about it here: http://pythoncentral.io/cutting-and-slicing-strings-in-python/

REFERENCES

Radio reference - or: all the stuff you can do with the radio module!
http://microbit-micropython.readthedocs.io/en/latest/radio.html

With material from readthedocs.io and Nicholas H.Tollervey.
Jessica Bergs // 28/01/2017, Craig Steele, Claire Quigley

have a look at this MicroBit Radio tutorial on
https://microbit-micropython.readthedocs.io/en/latest/tutorials/radio.html