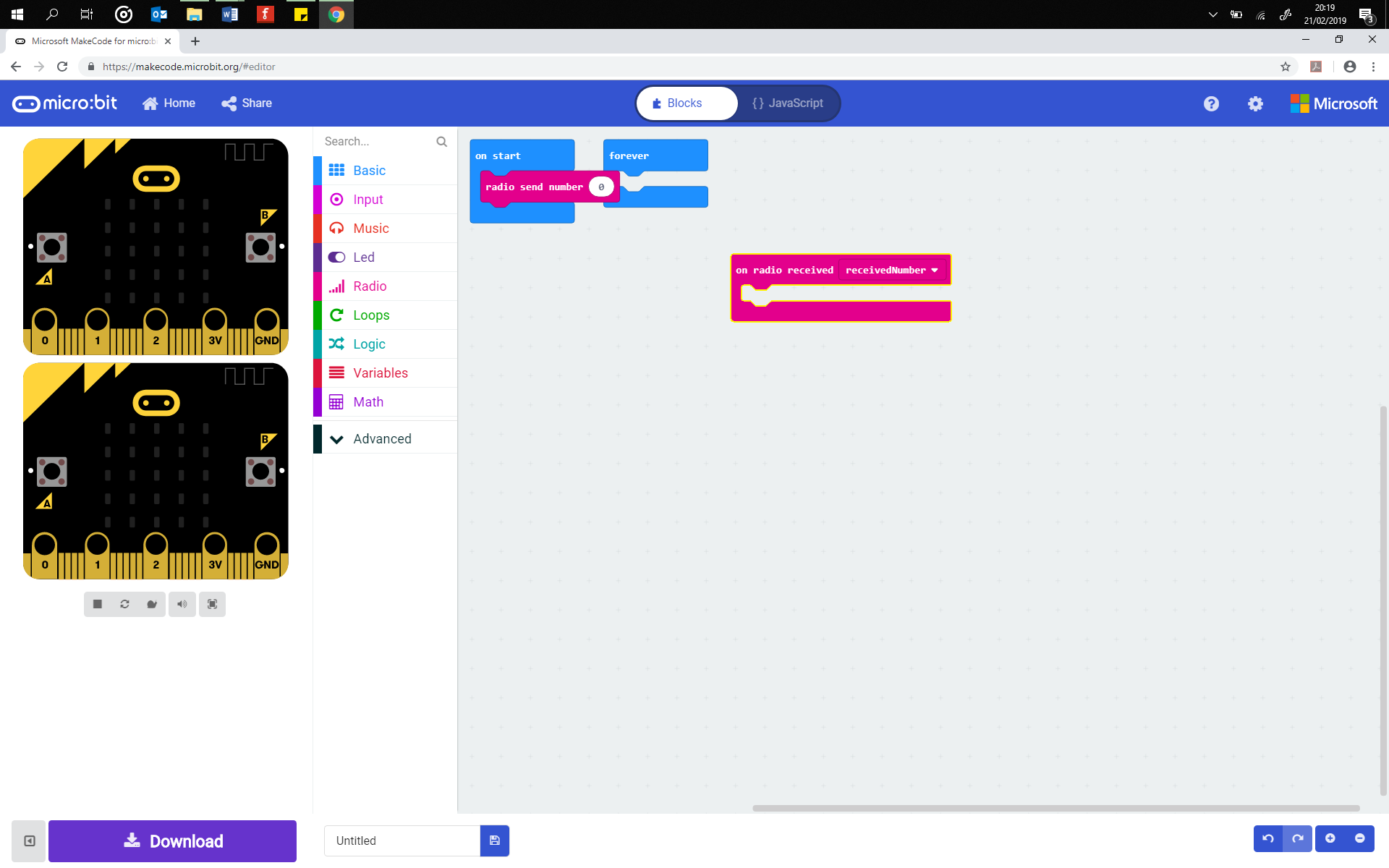
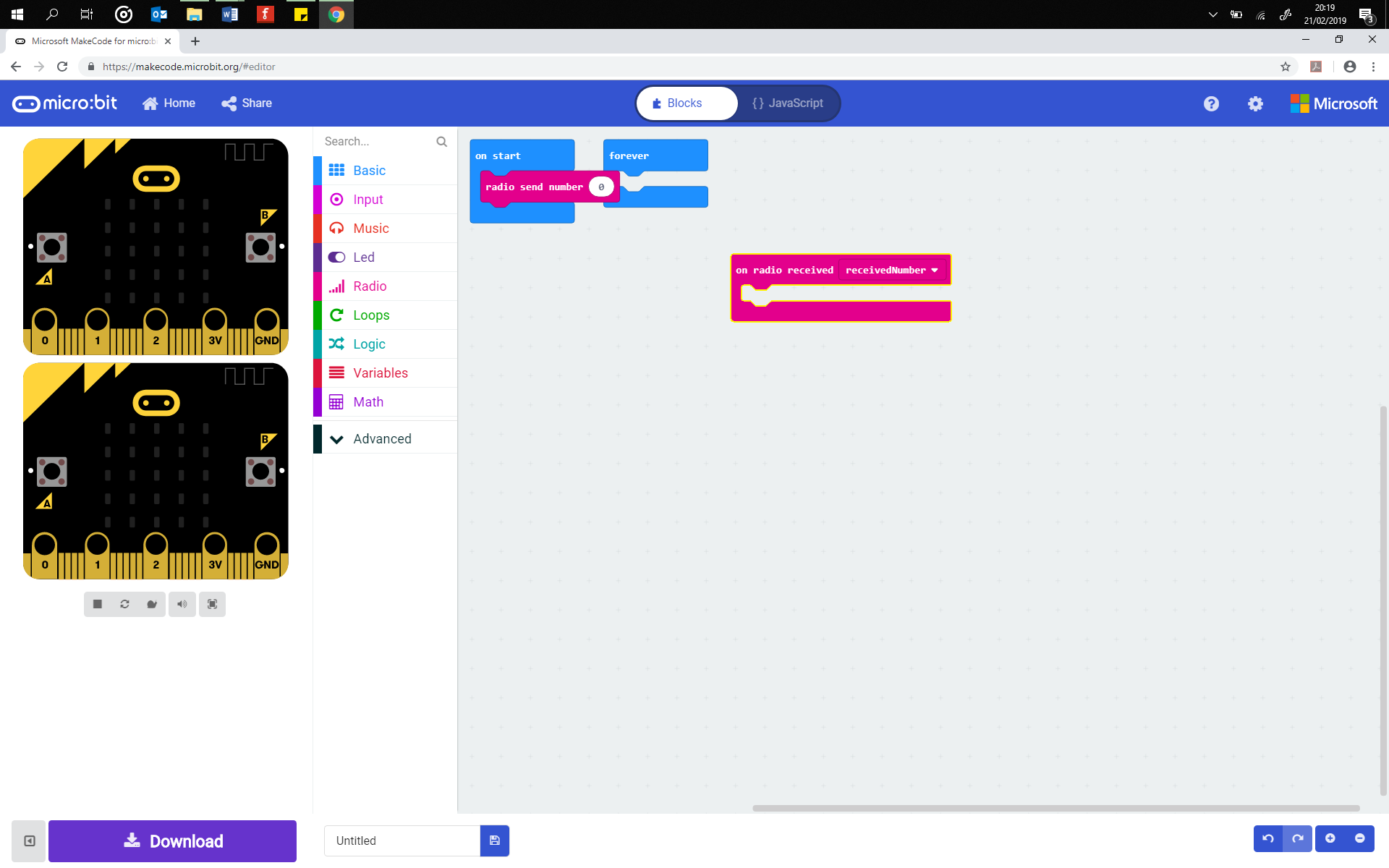
## Lesson 29 – Activity Sheet

Getting Stared

The radio module is great fun and very useful: you can send secret messages, control another micro:bit or create a light display that responds to the movement of another micro:bit.

**A Program to Send a Message**

This program requires two micro:bits. One is used as the sender and the other as the receiver. Pair up with another learner and decide who will be the sender and who will receive. The sender program asks the user to enter in a message which is stored in a variable. This message is then **broadcast** to another micro:bit. The receiver program is coded to check for and receive incoming messages. The program then responds by displaying the message that it has been sent from the other micro:bit.



Private message sent to micro:bit

## **Program 1: The Sender**

This program is responsible for sending a message via the radio signal from the micro:bit to another one. Decide which micro:bit the signal is sent to by using the radio.config() code, which you can set to a value between 0 and 100. This ensures that you transmit the signal to a particular micro:bit without anyone else receiving it. Change this from channel 5 to ensure that the message is not sent to all the micro:bits in the classroom. (*Don’t forget to tell your receiver what channel number you have picked*.)

The program starts by turning on the radio hardware and then setting the channel number. Next a while loop keeps the program looping so that you can send the message as many times as you want to. If you press Button A then the program sets the variable *my\_message* to the letter A. However, you can change this to a different letter, symbol or word. The radio then broadcasts the contents of the variable, your message to any micro:bits on the same channel. A short pause is used to add a delay before the button can be pressed again.

# Add your Python code here. E.g.

# radio 1

from microbit import \*

import radio

radio.on()

radio.config(channel=5)

while True:

if button\_a.was\_pressed():

my\_message = "A" Change the message to one of your choice

radio.send(my\_message)

sleep(200)

## **Program 2: The Receiver**

This is the receiver program which responds to the radio signal. It checks for incoming messages from another micro:bit, and then receives the message and displays it. Copy up the program code below and download to another micro:bit. Ensure that the same channel number is used as the receiver program.

# Add your Python code here. E.g.

# radio 2

from microbit import \*

import radio

radio.on()

# any channel from 0 to 100 can be used for privacy.

radio.config(channel=5)

This section of the code receives the incoming radio signal. Then the program checks if the incoming ‘is not’ None. This is a double negative which means the program checks if there is an incoming message. If there is then it uses the code display.show() to display the message that has been transmitted on the LED matrix.

while True:

incoming = radio.receive()

if incoming is not None:

display.show(incoming)

print(incoming)

sleep(500)

display.clear()

## Success Criteria

* Write up the sender program on one micro:bit
* Write up the receiver program on another micro:bit
* Successfully send a message between two micro:bits

## Pro-tip

Check that you are using the : symbol at the end of If statements and the while loops. Once downloaded and correct, the receiver program can stay the same whilst the sender program is added to or changed.

## Test Time!

Download the program codes to the micro:bits. Remember that the receiver program needs to be downloaded to one micro:bit and the sender to a different micro:bit. Ensure that both programs use the same channel number. This can be set between 0 and 100 but should be a different number from the channel number that the other learners are using.

## Stretch Tasks

## **Sending more than one message**

In this activity you will create a secret silent interview. One Learner will ask another questions. The Learner will use the buttons on their micro:bit to reply. The answer will be displayed on the micro:bit that the Learner asking the questions has. No one else will know the answer.

This program uses the radio chip to send a response to another Learner. The previous program can be adapted. However, you will want to program more than one response. This requires the use of selection and elif statements. Try out the example sender code below, don’t forget the check you are using the same channel number as your partner, this ensures no one else knows your responses.

# Add your Python code here. E.g.

#radio 1

from microbit import \*

import radio

radio.on()

radio.config(channel=5)

while True:

if button\_a.was\_pressed():

The program uses selection to enable the learner to send one of two responses. If they press Button A, then Y or Yes is sent. If they press Button B, then N for No is sent.

my\_message = "Y"

radio.send(my\_message)

sleep(200)

elif button\_b.was\_pressed():

my\_message = "N"

radio.send(my\_message)

sleep(200)

The receiver program is the same as before:

# Add your Python code here. E.g.

# radio 2

from microbit import \*

import radio

radio.on()

# any channel from 0 to 100 can be used for privacy.

radio.config(channel=5)

while True:

incoming = radio.receive()

if incoming is not None:

display.show(incoming)

print(incoming)

sleep(500)

display.clear()

Try out some of these additional features:

* Swap roles, the learner who was the receiver is now interviewed
* Add additional elif statements to provide more responses. For example, pressing Button A and B together could send the message ‘not sure’. Shaking the micro:bit could send the message ‘I refuse to answer that question!’ (*Remember that you only need to edit the sender program)*
* Edit the program so that the receiver can send back a secret response

## Final Thoughts

The radio hardware can be used to trigger any response on another micro:bit, for example, press Button A and all the micro:bits use the speech module to say ‘hello’! The possibilities are limitless – begin to think about how you might use the radio in the next project, the micro:PET.