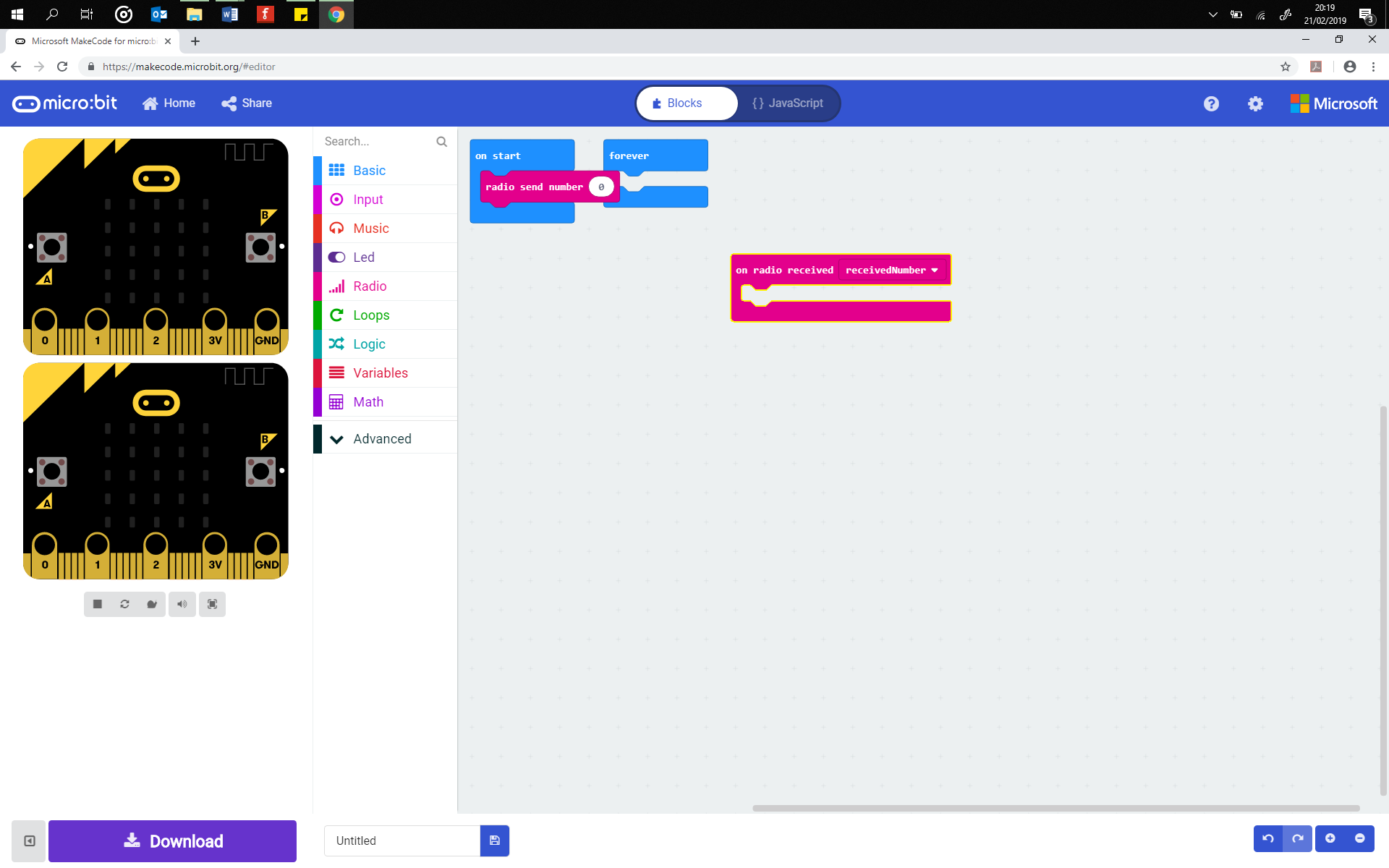
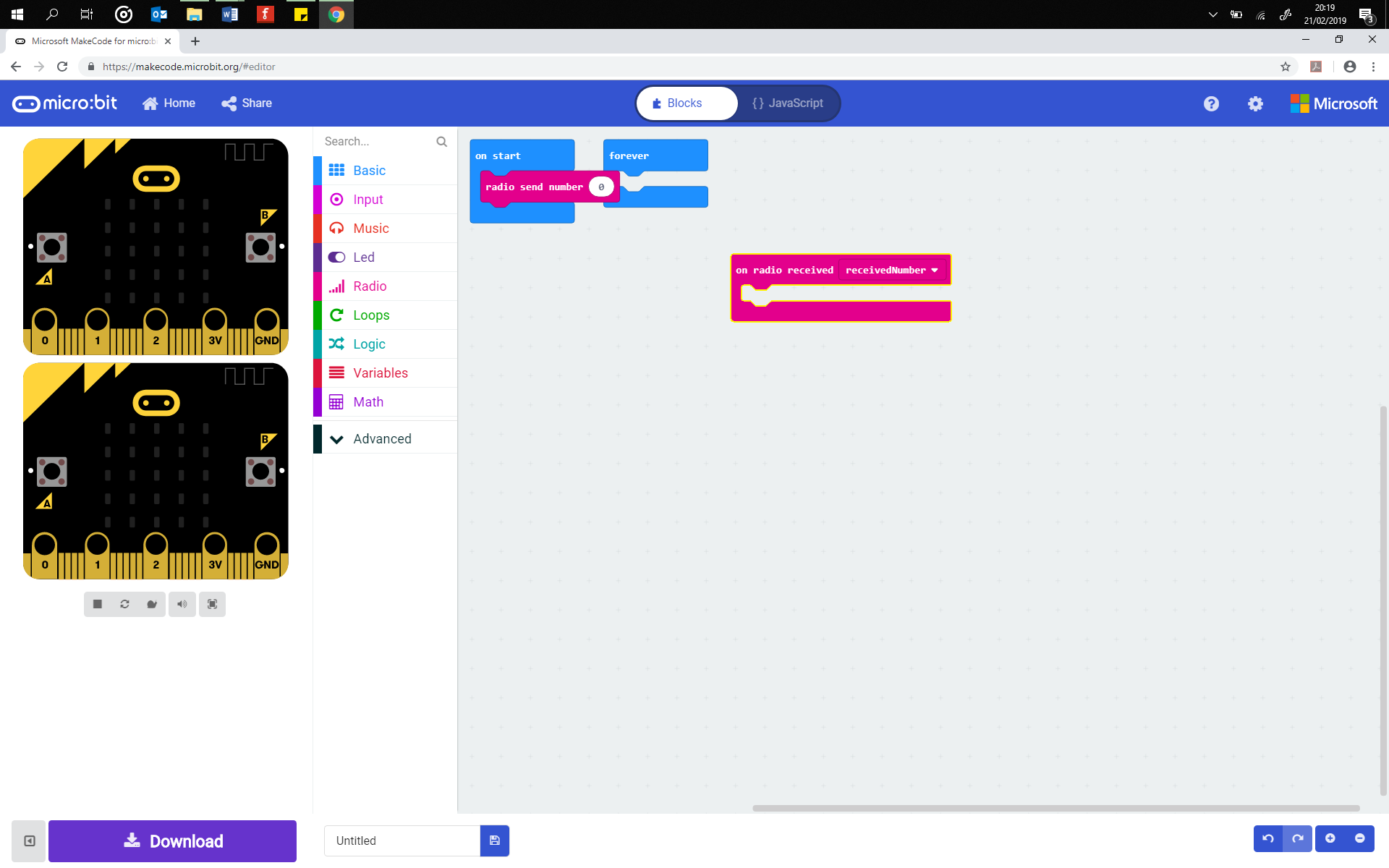
## Lesson 28 – Activity Sheet

## Getting Started

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

**Your first program**

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 broadcasts a trigger word to the other micro:bit using the radio module. The receiver program is coded to check for incoming messages on that channel and then respond to the trigger words. When the micro:bit receives the trigger word it responds by displaying either a happy face or a sad face.



Trigger word sent when Button A pressed

## **Program 1: The Sender**

This program is responsible for sending the radio signal from one micro:bit to another. 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 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 enables you to select Button A or Button B. If you press Button A then the radio sends the word ‘HAPPY’ to any micro:bit with the same channel number. If you press Button B then the word ‘SAD’ is sent to the other micro:bit. In your pairs decide who is going to be the sender and who is going to be the receiver. The sender writes up the program below. Then download to one of the micro:bits. If you are the receiver then look at program two.

# Add your Python code here. E.g.

# radio 1

from microbit import \*

import radio

radio.on()

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

radio.config(channel=5)

while True:

if button\_a.was\_pressed():

radio.send('HAPPY')

sleep(200)

elif button\_b.was\_pressed():

radio.send('SAD')

sleep(200)

## **Program 2: The Receiver**

This is the receiver program which responds to the radio signals that it receives from another micro:bit. The program is similar to the sender except that it defines how to respond when it receives the trigger words, ‘HAPPY’ or ‘SAD’. 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)

while True:

msg = radio.receive()

This section of the code receives the radio signal and stores it in a variable called msg. Then the program checks if the msg ‘is not’ (!=) None. This is a double negative which means the program checks if there is something stored in the msg variable. If there is, then it checks if it is the word, ‘HAPPY’. If it is then it displays the Happy image.

if msg != None:

if msg == 'HAPPY':

display.show(Image.HAPPY)

sleep(200)

display.clear()

elif msg == 'SAD':

display.show(Image.SAD)

sleep(200)

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 radio communication between two micro:bits
* Adjust the program (change the channel number, send a different message, use a different trigger)

## Pro-tip

Check that you are using the : symbol at the end of If statements and the while loops. Once download 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

* Change the channel numbers
* Swap roles, the learner who was the receiver tries out the sender program
* Use battery packs to power both micro:bits, test how far you can move apart from the sender and still send and receive the message
* Change the images
* Change when the event happens, for example on a shake, pressing buttons A and B together to send a third image
* Set the channel number to a random number and send the message, see if the receiver can locate you sent them the message

## Final Thoughts

In the next lesson we will look at how to send a message to another micro:bit – a secret message.