
The poster features a vibrant, multi-colored treble clef on the left. A micro:bit is shown with musical notes floating around it. The text 'Hack-a-tune' is written in a green, stylized font, with 'Code and create' in smaller green text below it. A white thought bubble contains the question: 'What's your favourite quote to inspire a positive learning environment?'. On the right, a white rounded rectangle contains the event title 'Hack - a - tune' in blue, followed by the description 'using micro:bits to explore the links between music, networking, cyber security and code' in black, and the time '11:40 – 12:15' in black. The bottom of the poster has a white banner with the 'FESTIVAL OF COMPUTING' logo, a circular logo with a cross and the text 'LEARNING POTENTIAL CREATING THE FUTURE', the event name 'Hack-a-Tune', the name 'Pete Dring', and the email 'dringp@fulford.york.sch.uk'.

Hack - a - tune

using micro:bits to explore the links between **music**, **networking**, **cyber security** and **code**


11:40 – 12:15

What's your favourite quote to inspire a positive learning environment?

FESTIVAL OF COMPUTING  **Hack-a-Tune** **Pete Dring** dringp@fulford.york.sch.uk

Welcome! For the first activity you'll need to think of a positive quote – perhaps that one cheesy line that you want to be remembered for saying all of the time. For me it's “you're worth infinitely more than your exam results”

The plan

FESTIVAL OF COMPUTING  **Hack-a-Tune** **Pete Dring** **dringp@fulford.york.sch.uk**

If you want or need any of the kit we're using today I'd strongly encourage you to apply for a meet and code grant: up to 3 x £400 = £1,200 grants:

<https://www.meet-and-code.org/gb/en/registration-info>

Image sources:

Microbit: <https://thepihut.com/products/micro-bit-v2>

AKAI APK Mini Play: <https://www.amazon.co.uk/AKAI-Professional-Mini-Play-built/dp/B09NQBDGT3>

Context

20

1853

8.9

6465

800

I've been teaching 20 years

I'm currently Head of Computing at Fulford School in York (non selective, mixed, state funded academy with 1853 students)

York is an affluent area (8.9% qualify for FSM) but with pockets of deprivation and almost the lowest per pupil funding in the UK (£175 below the government's £6640 minimum per pupil funding)

Which is why an £800 grant from the Digit<All> Foundation for a Meet and Code grant makes such a difference to run these sorts of activities for students – more details later about how you can apply for your own funding

Sources:

Meet and code: <https://www.meet-and-code.org/gb/en/>

Minimum funding levels:

https://assets.publishing.service.gov.uk/media/69417d442d5e7e863253755a/The_national_funding_formula_for_schools_and_high_needs_-_2026_to_2027.pdf


Deprivation maps: <https://deprivation.communities.gov.uk/maps>

Per pupil funding for each school: <https://explore-education-statistics.service.gov.uk/find-statistics/school-funding-statistics/2025-26/explore#data-guidance-section>

```
1 # Imports go at the top
2 from microbit import *
3
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7     display.show(Image.HEART)
8     sleep(1000)
9     display.scroll('Hello')
10
```

“python microbit editor”

Task 1: Replace ‘Hello’ with your inspiring quote

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk

Link to python microbit editor: <https://python.microbit.org/v/3>

All code available at: <https://github.com/pddring/MicroBitOrchestra>

Direct link to this code:

<https://github.com/pddring/MicroBitOrchestra/blob/master/micropython%20for%20microbit/01%20First%20challenge.py>

Image source: royalty free icon

```
# Imports go at the top
from microbit import *

# Code in a 'while True:' loop repeats forever
while True:
    display.show(Image.HEART)


    # show a play icon if you press button A
    if button_a.was_pressed():
        display.show(Image("09000:09900:09090:09900:09000"))

        sleep(500)

# Challenges
# 1) Fill in the dot in the middle of the play icon
# 2) Make a different image appear when you press the B button
# 3) Animate the play icon so it moves across the screen
```

Task 2:

Respond to button presses with an image or animation

FESTIVAL OF COMPUTING  **Hack-a-Tune** **Pete Dring** dringp@fulford.york.sch.uk

Link to python microbit editor: <https://python.microbit.org/v/3>

All code available at: <https://github.com/pddring/MicroBitOrchestra>

Direct link to this code:

<https://github.com/pddring/MicroBitOrchestra/blob/master/micropython%20for%20microbit/02%20Images%20and%20animation.py>

Image source: royalty free icon

```
# Imports go at the top
from microbit import *
import radio
radio.on()


# Code in a 'while True:' loop repeats forever
while True:
    display.clear()

    # show a play icon if you press button A
    if button_a.was_pressed():
        display.show(Image("09000:09900:09090:09900:09000"))
        radio.send("You are worth infinitely more than your exam results")
        sleep(500)

# Challenges
# 1) Send a different message when you press the B button
# 2) Change the play icon to a radio icon
# 3) Animate the radio icon so it looks like it's sending
```

Task 3:

Send your inspirational message using the radio

FESTIVAL OF COMPUTING  **Hack-a-Tune** **Pete Dring** dringp@fulford.york.sch.uk

Link to python microbit editor: <https://python.microbit.org/v/3>

All code available at: <https://github.com/pddring/MicroBitOrchestra>

Direct link to this code:

<https://github.com/pddring/MicroBitOrchestra/blob/master/micropython%20for%20microbit/03%20Sending%20radio%20message.py>

Image source: royalty free icon

```
# Imports go at the top
from microbit import *
import radio
radio.on()


# Code in a 'while True:' loop repeats forever
while True:
    # Show a line at the bottom of the screen to show it's listening
    display.show(Image("00000:00000:00000:00000:99999"))

    # message will be None if nothing has been received
    message = radio.receive()

    # display any received data on the REPL
    if message != None:
        print(message)
        display.show(Image("00900:99999:09990:00900:99999"))
    sleep(50)
```

Task 4 (Teacher only):

Display any received messages

FESTIVAL OF COMPUTING  **Hack-a-Tune** **Pete Dring** dringp@fulford.york.sch.uk

Link to python microbit editor: <https://python.microbit.org/v/3>

All code available at: <https://github.com/pddring/MicroBitOrchestra>

Direct link to this code:

<https://github.com/pddring/MicroBitOrchestra/blob/master/micropython%20for%20microbit/04%20Receiving%20radio%20messages.py>

Image source: royalty free icon

```
# Imports go at the top
from microbit import *
import radio
radio.on()

# Code in a 'while True:' loop repeats forever
while True:
    display.clear()


    # Send the number 76 if you press the A button
    if button_a.was_pressed():
        display.show(Image("09000:09900:09090:09900:09000"))
        radio.send("76")

    # Send the number 75 if you press the B button
    if button_b.was_pressed():
        display.show(Image("09000:09900:09090:09900:09000"))
        radio.send("75")

    sleep(500)
```

Task 5

Send MIDI note numbers

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk

Link to python microbit editor: <https://python.microbit.org/v/3>

All code available at: <https://github.com/pddring/MicroBitOrchestra>


Direct link to this code:

<https://github.com/pddring/MicroBitOrchestra/blob/master/micropython%20for%20microbit/05%20Sending%20MIDI%20numbers.py>

Image source: royalty free icon

Analogue Audio

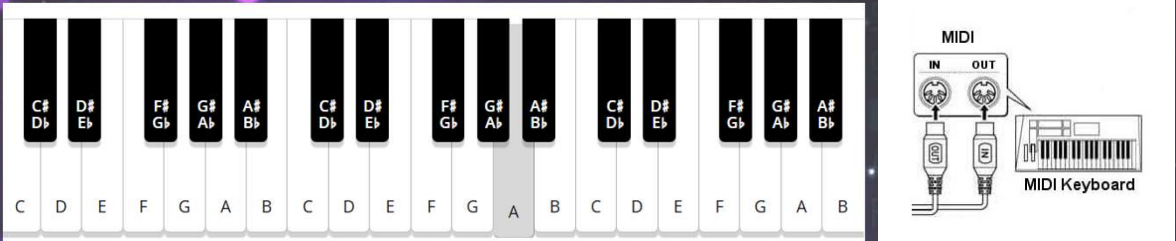
440Hz

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk

When you play the A key above middle C on a piano keyboard, the strings vibrate at 440Hz (if it's in tune!)

Sound wave: Audacity: <https://www.audacityteam.org/>

Virtual piano: <https://www.onlinepianist.com/virtual-piano>





Hex	9	0
Binary	1001	0000
Description	Note on	Channel 1

MIDI Note On

Hex: 90 45 7F

1001nnnn	0kkkkkkk	Note On event. This message is sent when a note is depressed (start). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity.
	0vvvvvvv	



Hack-a-Tune
Pete Dring
dringp@fulford.york.sch.uk

MIDI (Musical Instrument Digital Interface) uses binary to send messages that describe musical events (e.g. note on / note off). The 3 bytes on screen (0x90457F) represents the same 440Hz A above middle C

Virtual piano: <https://www.onlinepianist.com/virtual-piano>

MIDI specification: <https://midi.org/summary-of-midi-1-0-messages>

MIDI diagram: <https://www.icconnectivity.com/blog/2019/5/10/an-introduction-to-midi-connections>

Hex	4	5
Binary	0100	0101
Description	Note 69 (A above middle C)	

MIDI Note On
Hex: 90 45 7F

1001nnnn	0kkkkkkk Ovvvvvvv	Note On event. This message is sent when a note is depressed (start). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity.
----------	----------------------	--

FESTIVAL OF COMPUTING **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk

Virtual piano: <https://www.onlinepianist.com/virtual-piano>

MIDI specification: <https://midi.org/summary-of-midi-1-0-messages>


MIDI diagram: <https://www.icconnectivity.com/blog/2019/5/10/an-introduction-to-midi-connections>

Hex	7	F
Binary	0111	1111
Description	Maximum volume (127)	

MIDI Note On

Hex: 90 45 7F

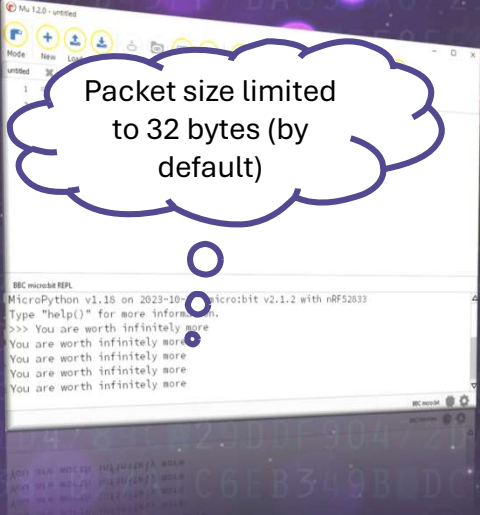
1001nnnn	0kkkkkkk	Note On event. This message is sent when a note is depressed (start). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity.
	0vvvvvvv	

FESTIVAL OF COMPUTING 
Hack-a-Tune Pete Dring dringp@fulford.york.sch.uk

Virtual piano: <https://www.onlinepianist.com/virtual-piano>


MIDI specification: <https://midi.org/summary-of-midi-1-0-messages>

MIDI diagram: <https://www.icconnectivity.com/blog/2019/5/10/an-introduction-to-midi-connections>



Concepts / next steps:

- **Packet size**
- Protocols
- DoS / DDoS
- Encryption
- Firewall
- Robotics

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk

The micro:bit uses micropython with a built in module for radio transmission. By default the packet size for sending data is limited to 32 bits so your inspirational messages may be truncated.

This is a great way to visualise the concept of splitting data into packets.


You can either increase the packet size in python or you can send separate parts of larger messages sequentially.

```
1 # Imports
2 from micro import radio
3 import time
4 radio.config(channel=10)
5
6 # Play the hi-hat if you press the A button
7 while True:
8     if button_a.was_pressed():
9         display.show("H")
10        radio.send("42 10")
11
12 # Play the snare if you press the B button
13 if button_b.was_pressed():
14     display.show("S")
15     radio.send("40 10")
16     sleep(100)
17
18 # Challenges:
19 # 1) Replace the numbers with your own percussive instrument
20 # https://musescore.org/sites/musescore.org/files/General127
```

Note number followed by channel
(10 is percussion)

Concepts / next steps:

- Packet size
- **Protocols**
- DoS / DDoS
- Encryption
- Firewall
- Robotics

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk

A communication protocol means an agreed set of rules that both sides follow in order to understand each other when transmitting and receiving messages.

As well as sending note numbers in MIDI we can send a channel. Channel 10 is percussion. The code above shows a hi hat on button a (note 42 on channel 10) and a snare (note 40 on channel 10).


If we want to send both the note number and channel we need to extend our radio protocol so that both sides know to send / receive two numbers (note: 1-127, then channel: 1-16)

How could you prevent this?!

```
1 # Import
2 from microbit import
3 import radio
4 radio.on()
5
6 # Code in a 'while True:' loop repeats forever
7 while True:
8     radio.send("!!!DoS!!!")
9
10     sleep(100)
11 |
```

Concepts / next steps:

- Packet size
- Protocols
- **DoS / DDoS**
- Encryption
- Firewall
- Robotics

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk



When you do this with a class it doesn't take long before someone sends the same message over and over again. This is an ideal opportunity to explain Denial of Service attacks: it's hard to see everyone's positive inspirational messages if someone is repeatedly sending the same message. This is a DoS. We can solve it by finding the annoying person (there's always at least one!) and unplugging their microbit

If everyone repeatedly sends repeated unwanted messages that's harder to block: that's a distributed denial of service

CyberChef

Concepts / next steps:

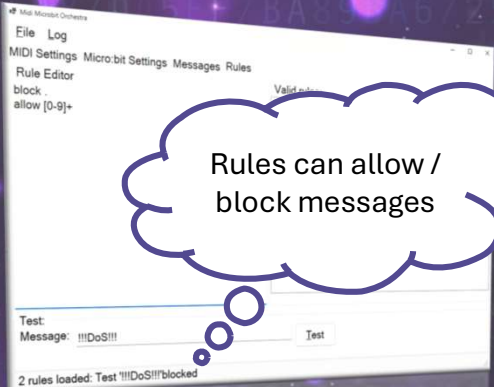
- Packet size
- Protocols
- DoS / DDoS
- **Encryption**
- Firewall
- Robotics



 Hack-a-Tune Pete Dring dringp@fulford.york.sch.uk

The microbit python module is public: anyone with a microbit can receive the data. Encryption is beyond the scope of this session but there's no reason why you couldn't teach students how to encrypt their messages using a Caesar cipher or XOR encryption.

The screenshot shows cyberchef which is a great tool for teaching cyber security / digital forensics related topics like encryption:


[https://gchq.github.io/CyberChef/#recipe=XOR\(%7B'option':'Hex',string:'C4'%7D,'Standard',false\)&input=NDlgMTA](https://gchq.github.io/CyberChef/#recipe=XOR(%7B'option':'Hex',string:'C4'%7D,'Standard',false)&input=NDlgMTA)



Rules can allow / block messages


Concepts / next steps:

- Packet size
- Protocols
- DoS / DDoS
- Encryption
- **Firewall**
- Robotics

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk



A firewall is just a set of rules which either allow or block network traffic. We can add rules to selectively allow messages we want to process and ignore everything else.

TCP/IP firewalls can filter by source / destination addresses as well as the message content but sadly we can't do that (unless we implemented our own source / destination addresses in our own protocol, but that leads to an interesting discussion about spoofing (faking) addresses



Concepts / next steps:

- Packet size
- Protocols
- DoS / DDoS
- Encryption
- Firewall
- **Robotics**

Hack-a-Tune

Pete Dring

dringp@fulford.york.sch.uk

We don't have time to demonstrate this today (and I couldn't carry all the kit on the train) but this is where the real fun starts. All the pictures are of cheap(ish) / donated radio controlled toys that we've taken to pieces and connected up to a microbit motor controller.


With the robot arm, you could get different students controlling different joints of the arm with their own microbit.

Similarly, the nerf shooting hexapod or the RC excavator are great for open evening when controlled by micro:bits

These motor controller boards are excellent: <https://thepihut.com/products/klip-motor-driver-for-bbc-micro-bit?variant=38151560528067>

Thank you

Any questions?

FESTIVAL OF COMPUTING  **Hack-a-Tune** Pete Dring dringp@fulford.york.sch.uk

I hope this has been useful: if you have any stories of devices you've taken to pieces and controlled with micro:bits I'd love to hear from you!