



Advanced Micro:bit Workshop



LEAVING CERTIFICATE
COMPUTER SCIENCE

Session 1 Part A Radio Communication & Serial Data Transfer.

Radio Communication & Serial Data Transfer

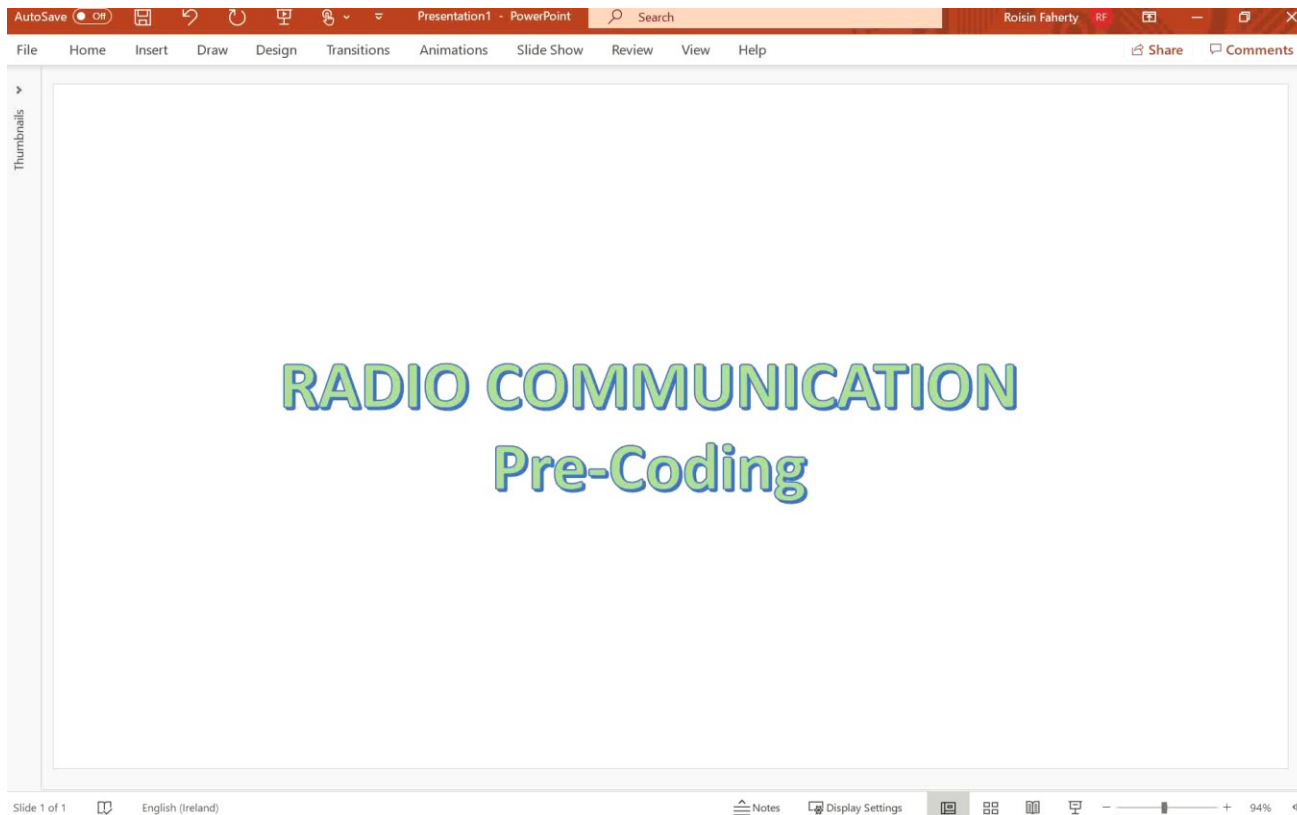
Aims of this section:

- To discuss how the radio communication on the micro:bit works.
- To illustrate the code for this using temperature data.
- Show how the received data can be graphed and downloaded as a CSV file.
- To use these ideas to complete a task on radio communication involving the viewing of received data and downloading this data to a CSV file.



Radio Communication

Pre- Coding (Design)



Radio Communication

Next Step is to code this on the micro:bit.

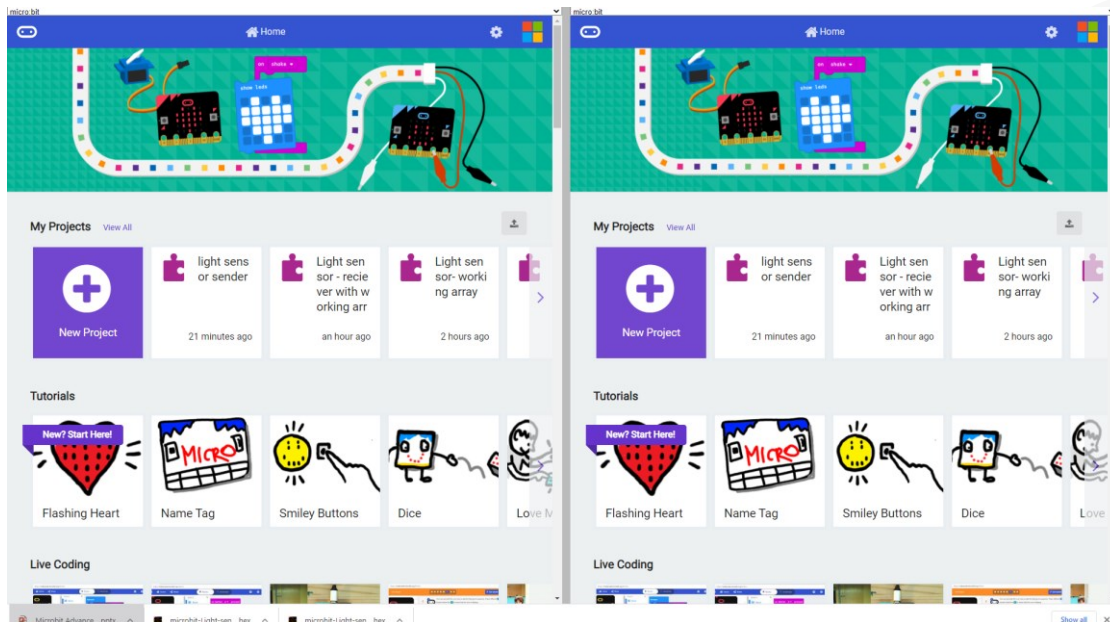
Click the link below:

[Make Code Multi Editor](https://makecode.com/multi#)

OR

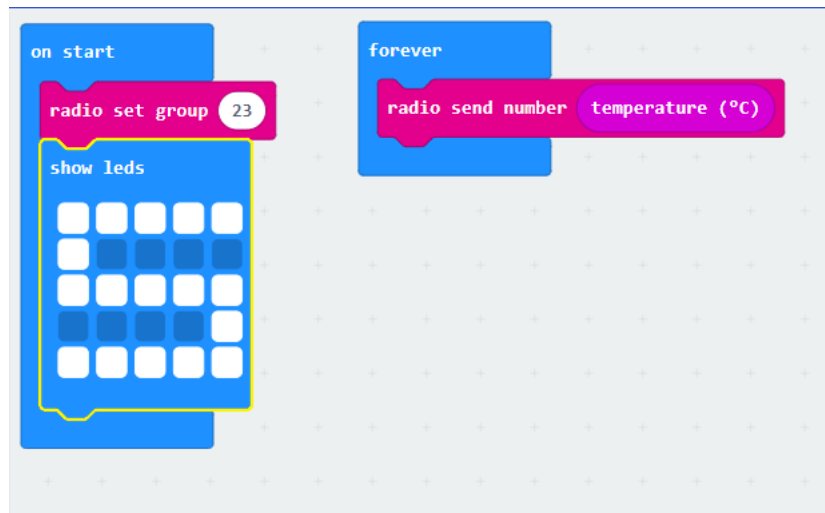
<https://makecode.com/multi#>

We will work through an example.

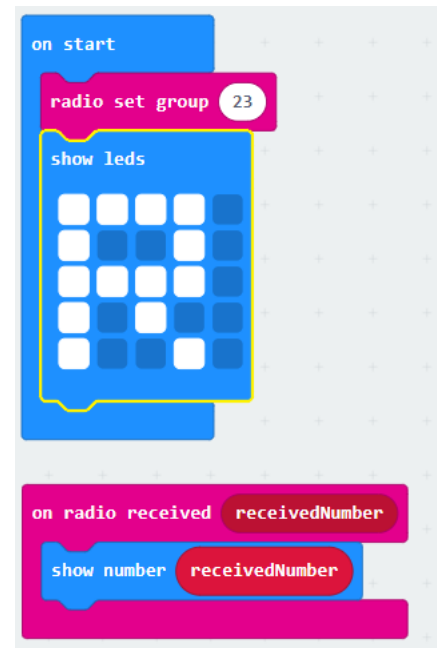


Radio Communication

SENDER CODE:



RECIEVER CODE:



Serial Communication

Pre-Coding (Design)

The image is a screenshot of a Microsoft PowerPoint application window. The title bar at the top shows 'Presentation1 - PowerPoint' and the user 'Roisin Faherty'. The ribbon menu includes 'File', 'Home', 'Insert', 'Draw', 'Design', 'Transitions', 'Animations', 'Slide Show', 'Review', 'View', and 'Help'. The main slide area contains the text 'SERIAL COMMUNICATION' in large, bold, green capital letters with a blue outline, and 'Pre-Coding' below it in a similar style. The status bar at the bottom indicates 'Slide 1 of 1', 'English (Ireland)', and a zoom level of '94%'. The 'Thumbnails' pane on the left is visible but empty.

Serial Communication

Next Step is to code this on the micro:bit.

Go To:

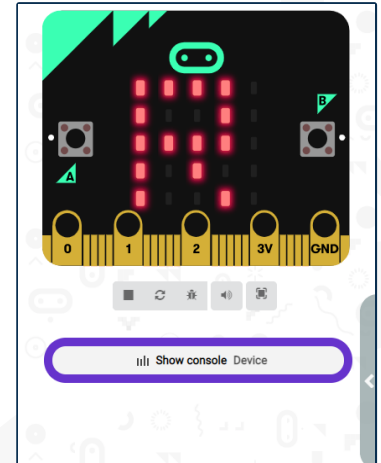
<https://makecode.com/>

Or

Download the offline micro:bit from the Microsoft site (more stable) at

<https://makecode.microbit.org/offline-app>

We will work through an example.



Group Task



TASKS

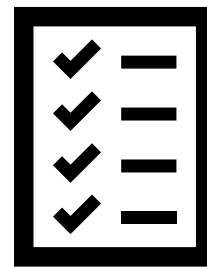
For this and all tasks today you will need:

Person to share screen for the coding and complete the code for the group.

A person to record the group Task Reflection.

A person who, on return to the main room, is going to report how the group got on.

Learning Activity Instructions



Based on the samples covered your task is to:

Part 1:

- Send light level via radio signal from one microbit to another.
- If the light level received is below 128 then get the receiver micro:bit to display that it is dark otherwise display it is bright. Consider the design of your display.
- Test to ensure this works on the virtual micro:bits.

Part 2:

- Open the receiver code in the normal micro:bit environment or in the offline version.
- Now update the code so that the received data is sent across the serial port.
- Don't forget to plug in and pair your device, then download the code!
- View the live data on the simulator and download the .csv file of data.
- Complete the group task Reflection Document.

TASK Solution Code

Part 1.

SENDER

The SENDER code consists of two main blocks:

- on start** (blue block):
 - radio set group** (pink block) with the value **23**.
 - show leds** (blue block) containing a 4x4 grid of 16 small blue squares.
- on button A pressed** (pink block):
 - radio send number** (pink block) with the value **light level**.

RECIEVER

The RECIEVER code consists of two main blocks:

- on start** (blue block):
 - radio set group** (pink block) with the value **23**.
 - show leds** (blue block) containing a 4x4 grid of 16 small blue squares.
- on radio received receivedNumber** (pink block):
 - if** (teal block) with the condition **receivedNumber ≤ 128** and the label **then**.
 - show string** (blue block) with the value **"D"**.
 - else** (teal block) with a minus sign icon.
 - show string** (blue block) with the value **"B"**.
 - +** (teal block) with a plus sign icon.

TASK Solution Code

Part 2.

This takes a few minutes to display but then allows us to track the live data.

The screenshot shows the Scratch IDE interface. On the left, a micro:bit is connected to a radio module. Below the micro:bit, there is a button labeled 'Show console Device'. On the right, the code editor shows the following blocks:

- on start** block containing:
 - serial redirect to USB** block
 - radio set group** block with value 23
 - show leds** block
- on radio received** block with variable **receivedNumber**:
 - if** block: **receivedNumber** \leq 128 **then**
 - show string** block with value "D"
 - else** block:
 - show string** block with value "B"
 - serial write line** block with variable **receivedNumber**

Added code to allow transmission via USB.

Added code to allow send the data.

TASK Solution Code

Visual of data tracking.

The screenshot displays the MakeCode for micro:bit IDE interface. On the left, a visual representation of the micro:bit board shows a 5x5 grid of red LEDs arranged in a pattern. Below the board is a 'Show console Device' button. The central workspace shows a script starting with 'on start' containing 'serial redirect to USB', 'radio set group 23', and a 'show leds' block. Below this, an 'on radio received receivedNumber' event triggers a sequence: 'show number receivedNumber', an 'if receivedNumber <= 128 then' conditional block, and 'serial write line receivedNumber'. The 'if' block contains 'show string "0"' and 'show string "1"' blocks. A 'Download' button is at the bottom left, and a 'Recliver solnpart1' button is at the bottom center.

MakeCode for micro:bit

micro:bit Home Share

Blocks JavaScript

Search...

Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
Advanced

on start

- serial redirect to USB
- radio set group 23
- show leds

on radio received receivedNumber

- show number receivedNumber
- if receivedNumber <= 128 then
 - show string "0"
- else
 - show string "1"
- serial write line receivedNumber

Show console Device

Download

Recliver solnpart1

Reflection



Stretch break

