Bit Radio

Stay Tuned

Alexandros Tasos
Aidan Oliver Thomas Hogg

Making radio stations and receivers from Microbits

THE CONCEPT

A simple idea:

- There's a number of radio stations (each with its own frequency) that broadcast their own Morse code transmission.
- Each one of these radio stations uses a Microbit in order to broadcast their transmission.
- Meanwhile, on the other side of the metaphorical fence, numerous Microbits that work as receivers tune into a specific radio station and output the station's signal via headphones or speakers.
- This idea is based on a DIY AM/FM radio kit.

BUILD ENVIRONMENT

Yotta: A module management system for C++ and C designed for Mbed OS, but you can use it on the Microbit too.

License: yotta is licensed under Apache-2.0

SENDING MESSAGES

- The Microbit is equipped with a radio module (not to be confused with the low range Bluetooth one).
- Each one of these radio stations uses a Microbit in order to broadcast their transmission.
- Although intended for one-to-one communication, the radio module API also allows us to broadcast a signal to multiple Microbits in proximity.
- The API also supports multiple frequency channels, which means that it will be possible for the end user to change the station they're tuned into.

INPUT/OUTPUT

Well, it's actually pretty simple:

The Microbit API allows us to read and write signals from/to the GPIO pins.

The API allows the emission and reception of digital (discrete) and the reception of analogue (continuous) signals.

PROCESS

First, we need to sample the ‘telegraph key’ input signal, which can be easily done through the API.

Then, we need to quantize the samples before transmitting them over the radio.

Finally, we need to play back the sound we receive from the radio station.

INTENDED LEARNING OUTCOMES

- To give young children an introduction to signals and communication.
- To demonstrate how a digital transmission system works.
- To get young children excited about electronics and the things it can accomplish.

CHALLENGES

- Unfortunately, the Microbit is less powerful compared to any 90s console sound chip (even that of the SNES).
- Maximum packet size that the radio module can send is 249 bytes.
- The GPIO pin API doesn’t let us use Pulse Code Modulation to play back the signals received, but instead it uses Pulse Width Modulation.
- This meant that we had to make compromises in our design and implementation.
- We originally indented for the device to perform real-time audio transmission, however, we had to settle for Morse code transmission due to the packet size limitation.

SUMMARY

The aim of this project was to develop some software on the Microbit to teach and excite kids.

I think we achieve this in our end product “Bit Radio” which should be fun and educational to the young user.

Future work: This concept could be taken further by using speech signals instead of Morse code.

ACKNOWLEDGEMENTS

We wish to thank various people for their contribution to this project:

Dr Ben Glocker (Imperial College London)

Lee Stott (Microsoft)

Peli de Halleux (Microsoft)

A special thanks should be given to Microsoft and the HIPEDS CDT for sponsoring this project.

REFERENCES

Yotta: lancaster-university.github.io/microbit-docs/offline-toolchains/#yotta

Mircobit website: microbit.co.uk