The BBC micro:bit - from the UK to the World

[Extended Abstract]

Lancaster University

Micro:bit Educational Foundation

Microsoft

ABSTRACT

The micro:bit rocks!

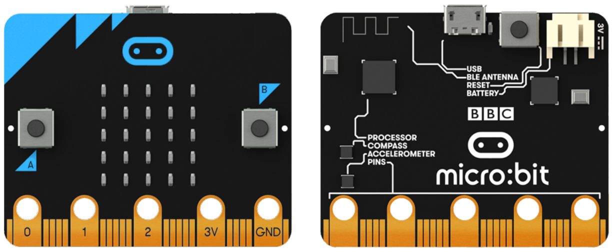
1. INTRODUCTION

The micro:bit is a small programmable and embeddable computer designed, developed and deployed by the BBC and partners (including ARM, Microsoft and Lancaster University) to approximately 800,000 UK middle school students in 2015-2016. The BBC micro:bit follows in the tradition of the BBC Micro personal computer from the 1980s; it also supports the UK's mandate to teach computer science concepts at all grade levels. [?] Part of the BBC's Make It Digital Campaign, the BBC described the micro:bit as its "most ambitious education initiative in 30 years, with an ambition to inspire digital creativity and develop a new generation of tech pioneers." [?]

Figure 1 shows (a) the front and (b) the back of the micro:bit, which measures 4cm x 5cm. Like the Arduino Uno, the micro:bit is a single-board microcontroller that can be programmed via a host computer (usually a laptop or desktop) and then embedded in projects where it runs on battery power. In contrast to the Uno, which has no built-in sensors, the micro:bit board hosts a variety of sensors (temperature, accelerometer, magnetometer, light level), a 5x5 LED matrix, two user-defined buttons, as well as Bluetooth Low Energy (BLE) communications.[1].

The design of the micro:bit hardware was driven by the first two objectives of the BBC micro:bit project: (B1) to provide a simple creative experience for physical computing, wearable and Internet of Things (IoT) projects; (B2) to supply a device that can continue to provide learning opportunities as the user's expertise grows.

On the hardware side, the micro:bit's built-in sensors, buttons and LED display allow many projects to be completed with no additional hardware or wiring. The holes on micro:bit's edge connector allows additional external sensors and actuators to be connected via crocodile clips. The micro:bit's BLE capabilities introduces networking to the picture, and enables streaming of data and command/control operations among the micro:bit, smartphones, laptops, as well as other micro:bits. As with Arduino, an ecosytem of micro:bit shields (hardware peripherals) that accommodate the micro:bit's edge connector expands its capabilities greatly.[2]



(a) (b)

Figure 1: The micro:bit: (a) front, with two buttons, 5x5 LED display, and edge connector (bottom); (b) back, with processor, accelerometer, compass, Bluetooth, USB and battery ports.

The design of the micro:bit coding tools also was oriented towards a simple starting experience with room for progression. In particular, the coding objectives of the project were: (B3) to give students an exciting, engaging introduction to coding; (B4) to stimulate curiosity about how computing technologies can be utilized to solve problems that students identify.

Based on in-school trials with a micro:bit prototype, the BBC focused on delivering a web app based on the popular Blockly framework [1] to permit students to create scripts via drag-and-drop operations in a web browser, and see the execution of their scripts via a simulator. Text-based coding via scripting languages also was identified as an important feature. As the micro:bit would be incorporated into standalone projects, it also was essential for the user's program to be compiled and installed in non-volatile storage on the micro:bit where it could be run via battery power.

The solution delivered by the BBC's partners evolved from the initial design to include:

- support for Blockly, JavaScript, Python and C++;
- an efficient C++ runtime for the micro:bit created by Lancaster University;

[1] The micro:bit has a whopping 16kB of RAM and 256kB of Flash memory, compared to the Uno's 2kB of RAM and 32kB of Flash

[2] `http://microbit.org/assets/documents/AccessoryGuideSummer18.pdf`

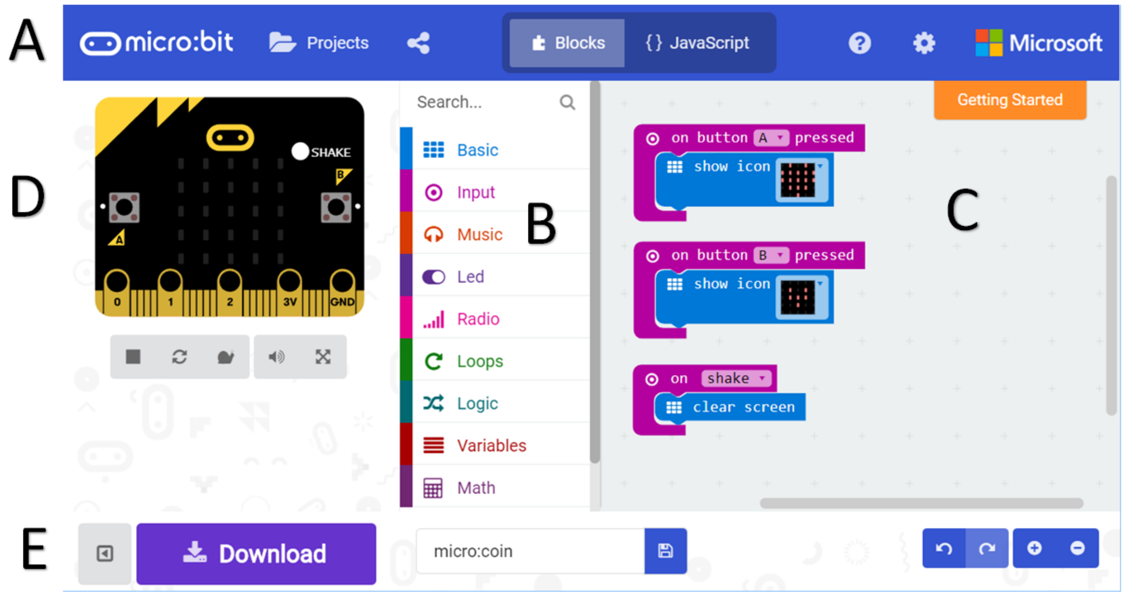


Figure 2: MakeCode web app for the micro:bit

- a web app (`http://makecode.microbit.org`) with Blockly and JavaScript editors, micro:bit simulator, and a compiler to machine code (linked against a pre-compiled C++ runtime image, so no C++ compiler is needed for user code);
- a Python compiler and read-eval-print loop (REPL) that resides *on the micro:bit* (via `https://micropython.org/`), supported by a simple web app (`http://python.microbit.org`) and an installable application (`https://codewith.mu/`);
- ARM's DAPlink firmware makes the micro:bit appear as USB pen drive on most operating systems, enabling a simple file copy operation to install a user's program on the micro:bit (no device drivers needed).

MakeCode, MicroPython, and the C++ runtime are all open source.[3]

Figure 2 shows a screen snapshot of the MakeCode web app for the micro:bit with five main sections: (A) menu bar with access to projects and examples, and switching between Blockly and JavaScript editors; (B) Blockly toolbox of micro:bit API categories; (C) Blockly programminng canvas with a simple reactive program; (D) micro:bit simulator for execution of the user's program in browser; (E) download button, which invokes the in-browser compiler/linker to produce a binary executable.

The event-based program shown in section (C) displays a large heart when the A button is pressed, displays a small heart when button B is pressed, and clears the display when the user shakes the micro:bit (shake detection is implemented using the accelerometer; in the simulator, the shake event is fired using a virtual button). In addition to event-based APIs, direct access to the micro:bit's sensors via polling is possible. [takes a few minutes to code and deploy this simple program]

The BBC micro:bit project also called for partners to develop content and to "train the trainers" (educators) around the micro:bit computing system. This built on efforts by the non-profit Computing At Schools (`www.cas.org`) organization to train UK educators to teach computer science.

Together, we learned many lessons from being a core part of the team that delivered the BBC micro:bit in the UK and then expanded to reach more educators and students around the world, via the Micro:bit Educational Foundation (`www.microbit.org`; started in September 2016) and its founding partners' efforts. In the remainder of this paper, we focus on the primary promise of the BBC micro:bit, which was to deliver a simple physical computing experience, en masse, for beginners and a progression path for users to follow as they gain knowledge and confidence. We draw from two full years of full deployment of the micro:bit in the UK, as well as deployments in Europe, the Americas, and Asia. There are approximately two million micro:bits now in the market and many hardware, content, and education partners participating.

We take a bottom-up approach, starting with a review of physical computing (Section 2), which anchors and defines the micro:bit experience. Section 3 presents a broad set of micro:bit-based projects, ranging from wearable games to environment science, that demonstrate the micro:bit's capabilities. Section 4 reviews the history of the BBC and computing, the rollout of the micro:bit in the UK and the establishment of the Micro:bit Educational Foundation. Section 5 describes the rollout of the micro:bit to other countries in Europe, the Americas and Asia. Section 6 concludes with final thoughts about what has made the micro:bit successful to date and what comes next.

2. CONTEXT: PHYSICAL COMPUTING

[3] At `https://github.com/microsoft/pxt`, `https://github.com/micropython/micropython`, and `https://github.com/lancaster-university/microbit-dal`, respectively.

As discussed in the Introduction, the micro:bit is a device with similarities to the Arduino family of printed circuit boards. Such *physical computing* devices are designed to be placed in and interact with our physical environment. Physical computing lives in the spaces between computing and many other disciplines: art, industrial design, health, environmental monitoring; it has close ties to cyber-physical systems, embedded systems, and IoT. The National Science Foundation defines cyber-physical systems as those that "integrate sensing, computation, control and networking into physical objects and infrastructure, connecting them to the Internet and to each other."[2]

The benefits of using physical computing to introduce beginners to computing systems include:

- *broad reach* because of diverse applications of physical computing – leverage fine arts, music, design, etc. in projects;
- *increased motivation* because of tangible visible outcome (rather than virtual on screen);
- *learning by doing* as there are many ways to achieve goal (no single correct solution)
- *natural division of labor* for more complex projects (design, hardware, software, ...)
- *full system view of computing*: hardware and software working together.

2.1 Wiring and Arduino

To help explain the BBC micro:bit, it's very instructive to understand Hernando Barragan's 2003 Master's thesis, "Wiring: Prototyping Physical Interaction Design", the inspiration for the Arduino system [**?**]. His objective was to make it easier for non-technical creators, such as artists and designers, to leverage electronics in their their work by simplifying the hardware and programming experience. In particular, he said of existing work: "Current prototyping tools for electronics and programming are mostly targeted to engineering, robotics and technical audiences." Of Wiring's design, he identified the following key concepts:

- a simple cross-platform integrated development environment (IDE) to create so-called "sketches";
- simplified application programming interfaces (APIs) to access a microcontroller's resources;
- leverage open source compiler/linker toolchain, transparent to the end user;
- a bootloader to make it easy to upload a compiled sketch to the microcontroller;

Also key to Wiring was openness of both the hardware and software comprising the system.

But, still some issues:

- reliance on the C language and C compiler (needs to be installed)
- very poor experience in IDE
- USB bootloader requires device drivers on some systems

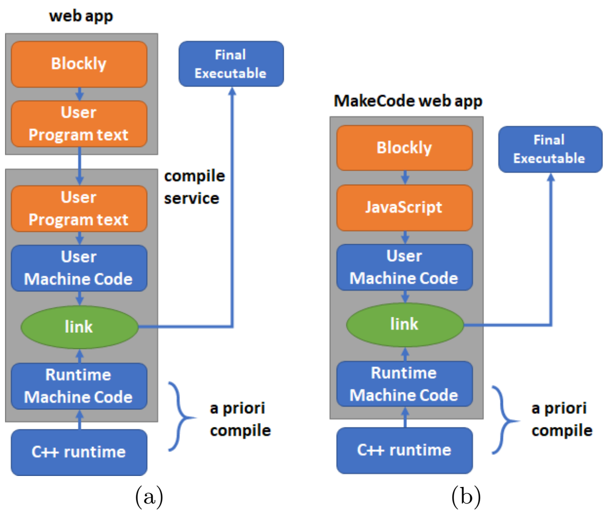


Figure 3: Web and compiler designs: (a) initial BBC design; (b) final design, as implemented in MakeCode.

2.2 The BBC micro:bit

Main points:

- *A Visible Computer*: BBC micro:bit inherits the raw PCB nature of Arduino (everything is visible to the end user).
- *No Wiring*: makes starting easy
- *Small Size*:
- *Scripting via Web App*: XYZ
- *No Install*: XYZ As shown in Figure 3(a), in the BBC design the text of a user's program (whether derived from Blockly or produced directly by the user) is submitted to a compile service that returns a final executable to be copied onto a micro:bit (connected to the host computer by USB) via a specialized loader application. avoiding the need for a compile service for user code (as shown in Figure 3(b));
- *Extensible*: via edge connector and layered APIS (package system too).

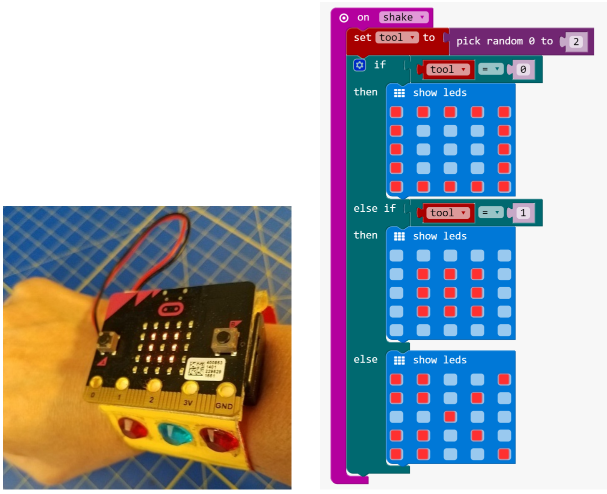
From this perspective, the micro:bit can be seen as a starter device for physical computing, embedded systems and cyber-physical systems, as it has sensing, computation, control and networking capabilities built in. The micro:bit is not properly an IoT device, having no built-in way to connect over IP, but it can be connected to other devices with IP connectivity.

3. PROJECTS

In this section we present a sampling of projects that illustrate the capabilities of the micro:bit.

3.1 Wear and Play

Figure 4 shows one of the most popular micro:bit projects: a watch that plays the rock/paper/scissors game when shaken; the program reacts to a shake event by choosing a random



(a) (b)

Figure 4: Micro:bit watch for playing rock/paper/scissors.

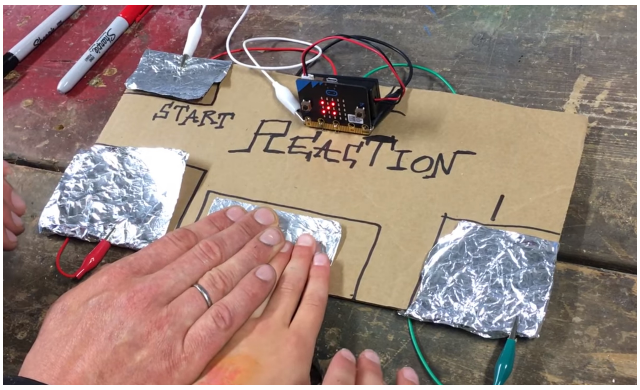


Figure 5: Reaction game.

integer between 0 and 2 and displaying a rock, paper or scissor shape on the LED display, based on the number chosen. The user can use this simple app to play the game with themselves or a a friend. The project consists of a making step and coding step, as shown at

`makecode.microbit.org/projects/rock-paper-scissors`

Many micro:bit projects use simple classroom supplies. The reaction game project (Figure 5) uses cardboard, aluminum foil, and crocodile clip connectors to illustrate the use of circuits with a game that measure reaction time. Crocodile clips connected to pins P0, P1, P2 and GND also are connected to aluminum pads. The user completes a circuit by touching the GND pad and one of other pads. The pad labelled “START” begins the game; after a 1-3 seconds (randomly determined), the micro:bit display lights up - the first user to touch their pad wins, and their reaction time is displayed:

`makecode.microbit.org/projects/reaction-time`

3.2 Measure

The micro:bit’s built-in sensors and small size make it perfect for embedding in science and technology projects. The

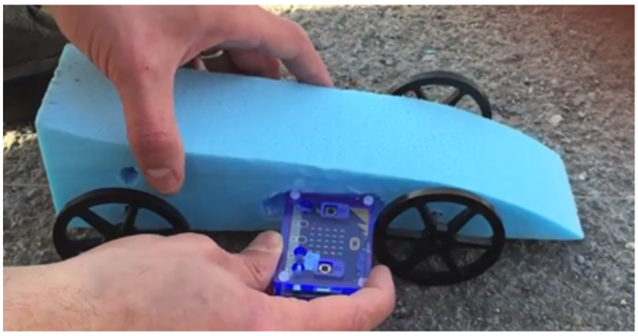


Figure 6: Bloodhound Model Rocket Car with embedded micro:bit for measuring acceleration.

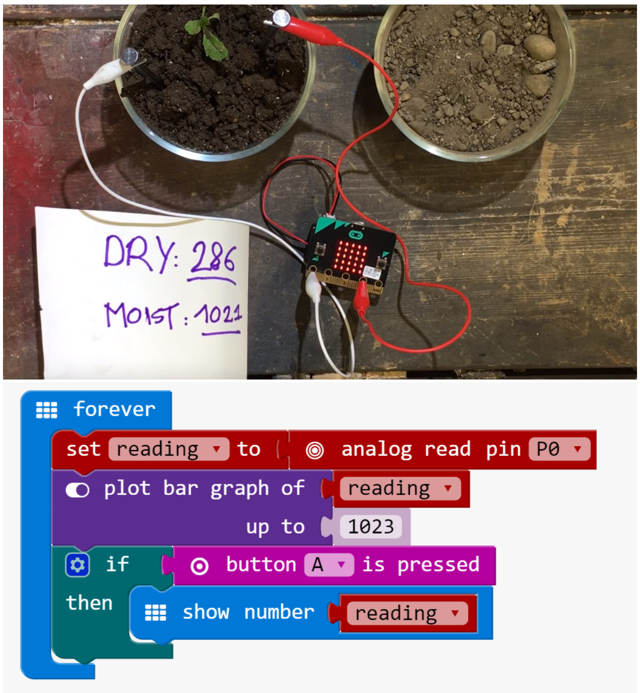


Figure 7: Measuring soil moisture via micro:bit pins.

Bloodhound Model Rocket Car is part of the Bloodhound Project, [4] whose goal is to set a new world land speed record and inspire students about STEM subjects. Students design, build and race model rocket cars in competition, learning about physics, aerodynamics, and mechanical engineering. Microsoft worked with the Bloodhound Project to incorporate a micro:bit into the car’s design, as shown in Figure 6; the micro:bit captures the (X,Y,Z) accelerometer data of the rocket car during its race. After the race, students can upload the data from the micro:bit and analyze the performance of their cars.

Figure 7 shows an environmental project that uses the micro:bit to measure soil moisture. The combination of water and soil nutrients makes the soil have some conductivity. The more water there is in the soil, the greater its conductivity, which can be measured using the analog pin API. In this project, the student first learns to calibrate the mea-

[4] `www.bloodhoundssc.com`

```
input.onButtonPressed(Button.A, () => {
  radio.sendString("H");
});

input.onButtonPressed(Button.B, () => {
  radio.sendString("S");
});

radio.onDataReceived(() => {
  let data = radio.receiveString();
  if (data == "H") {
    basic.showIcon(IconNames.Happy)
  } else if (data == "S") {
    basic.showIcon(IconNames.Sad)
  } else {
    basic.showString("?");
  }
});
```

Figure 8: . Broadcasting simple messages using the micro:bit radio.

surement readings using dry and wet soil samples. Then, the micro:bit is coded to periodically record the reading. Using the micro:bit's Bluetooth radio, the readings also can be sent to a central source. In this way, the moisture of a set of soil samples (in a classroom) can be recorded and reported. For more about this project, see:

`makecode.microbit.org/projects/soil-moisture`

3.3 Network

Using a lower level of the Bluetooth stack, the micro:bit supports a simple radio broadcast protocol that can be used to send short messages to a set of micro:bits. Figure 8 presents a simple example in JavaScript that shows how to use a micro:bit to communicate your "mood" to other micro:bits in the vicinity. Note that the micro:bit that sends a message does not receive that message.

The following two projects use the micro:bit radio to illustrate how fireflies synchronize their blinking and how infections spread:

`makecode.microbit.org/projects/fireflies`

`makecode.microbit.org/projects/infection`

3.4 Control

The micro:bit can be attached to external actuators, such as servos, to create systems that respond physically to their environment.

4. THE BBC AND THE FOUNDATION

The BBC's history with computing and education goes back to the early 1980's and the BBC Computer Literacy Project, which featured the BBC Micro, a 6502-based microcomputer designed and produced by Acorn Computers Ltd. (referred to at times as the "British Apple"). The project was very sucessful: more than 80% of UK classrooms had a BBC Micro and many of today's computing professionals from the UK first encountered computing through the BBC Micro.

In December 2014, the BBC issued an Request for Participation for "Delivery of a hands-on learning experience for the Make it Digital season", which was the micro:bit project. Twenty-nine partners were invited to contribute hardware, software, services, teaching materials, packing/distribution, logistics, events and funding. Work on the project commenced in February 2015, with delivery of a web site/app in September 2015 (which was critical for training teachers) and delivery of the micro:bits in the second half of the 2015-2016 school year.

Founded in September 2016, the Micro:bit Educational Foundation is a non-profit organization legally established with the support of its founding partners [5]. The Foundation's Mission Statement is to:

- enable and inspire all children to participate in the digital world, with particular focus on girls and those from disadvantaged groups.
- make micro:bit the easiest and most effective learning tool for digital skills and creativity.
- work in collaboration with educators to create and curate exceptional curriculum materials, training programmes and resources.
- build and support communities of educators and partners to remove the barriers to learning digital skills

The Foundation works to make micro:bits available for purchase (singly and in bulk) around the world through resellers. [6] The Foundation redistributes the bulk of any surplus money generated into providing free devices to exceptional micro:bit educational programmes across the globe.

5. COUNTRY DEPLOYMENTS

localization - language but also curriculum standards

6. CONCLUSION

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] N. Fraser. Ten things we've learned from blockly. In *Proceedings of the 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, BLOCKS AND BEYOND '15, pages 49–50, 2015.

[2] NSF. Cyber-physical systems. 2018.

[5] ARM, Amazon, BBC, British Council, IET, Lancaster University, Microsoft, Nominet, and Samsung

[6] Currently in Australia, Belgium, Brazil, Canada, China, Croatia, Czech Republic, Denmark, Estonia, Finland, France, Germany, Hong Kong, Hungary, India, Ireland, Israel, Italy, Japan, Latvia, Lithuania, Luxembourg, Malaysia, Netherlands, New Zealand, Norway, Poland, Singapore, Slovak Republic, South Africa, Spain, Sweden, Switzerland, Taiwan, Thailand, UK, and the US.