DOI:10.1145/3368856

A codable computer half the size of a credit card is inspiring students worldwide to develop core computing skills in fun and creative ways.

BY JONNY AUSTIN, HOWARD BAKER, THOMAS BALL, JAMES DEVINE, JOE FINNEY, PELI DE HALLEUX, STEVE HODGES, MICHAŁ MOSKAL, AND GARETH STOCKDALE

The BBC micro:bit—From the U.K. to the World

IN 2015, THE BBC launched the Make It Digital initiative, aiming to encourage a new era of creativity in the young using programming and digital technology as its medium. Simultaneously, the initiative also would support the U.K.'s mandate to teach computer science concepts at all grade levels.[13]

The micro:bit is a small programmable and embeddable computer designed, developed, and deployed by the BBC and 29 project partners to approximately 800,000 U.K. Year 7 (11/12-year-old) school children in 2015–2016. Referring back to its work with the BBC Micro,[4] 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."[1]

Embracing a constructionist approach to computing education,[11] the micro:bit has moved from a local educational experiment in the U.K. to a global effort driven by the Micro:bit Educational Foundation (microbit.org), a nonprofit organization established in September 2016. There are now over four million micro:bits in the market in over 60 countries with many hardware, content, and education partners participating.

The BBC and its partners developed the micro:bit as an inexpensive, powerful, and easy-to-use learning tool guided by five major design goals:

1. *Have a low barrier to entry.* Financial cost and simplicity are important considerations for any technology, but even more so in an educational setting. The micro:bit needed to be affordable, easy to deploy, intuitive to use, simple to program, and integrate well with existing school IT infrastructure.

2. *Be fun and creative.* The micro:bit itself needed to offer an exciting, engaging, inclusive introduction to coding and making. Inspired by Arduino and the Maker movement,[7] the project sought to turn teachers and students from digital consumers into digital creators by integrating the micro:bit into their own real-world, physical creations.

3. *Have a low floor, high ceiling, and wide walls.* When designing the micro:bit, providing good educational value to students and teachers was the prime consideration. It needed to be easy for inexperienced learners to get started (low floor); enable rich learning opportunities that grow with user expertise, provide progression in both programming language and application complexity (high ceiling); and enable students to reach the ceiling via multiple pathways to embrace a diverse audience (wide walls).[11,15]

4. *Open a window into the future.* Computing technology is becoming ever more ubiquitous, connected, and embedded. In the 1980s, the BBC Micro[4] captured the essence of the devices that were to come over the next

Girlsday, hosted by Microsoft, The Netherlands, drew many happy participants.

30 years: the desktop PC. The micro:bit was designed as a modern-day equivalent, capturing the connected, embedded nature of devices that are to come for the *next 30 years*.

5. *Be applicable beyond computer science.* Cross-curricular activities can offer diverse and inclusive learning.[3,12,16] This is important when we consider the gender disparity in computing today. The micro:bit project aimed to stimulate curiosity about how computing can be applied across a variety of disciplines, ranging from science and technology/engineering to the arts and mathematics (STEAM).

In this article, we describe the design of the BBC micro:bit and the realization of these goals, exemplified through a sample set of diverse projects. We review the project's history as it transitioned from a U.K.-centric to a worldwide project, concluding with lessons learned and project outcomes.

The BBC micro:bit

The BBC spent two years investigating previous work and new ideas to get more children coding and to improve digital literacy. Research shows that physical computing—combining software and hardware to build interactive physical systems that sense and respond to the real world—can engage a diverse range of students.[10] The simultaneous global interest in the maker movement also suggests an appealing way to engage children is to incorporate making, creating, and inventing as part of the software development process.[7,9]

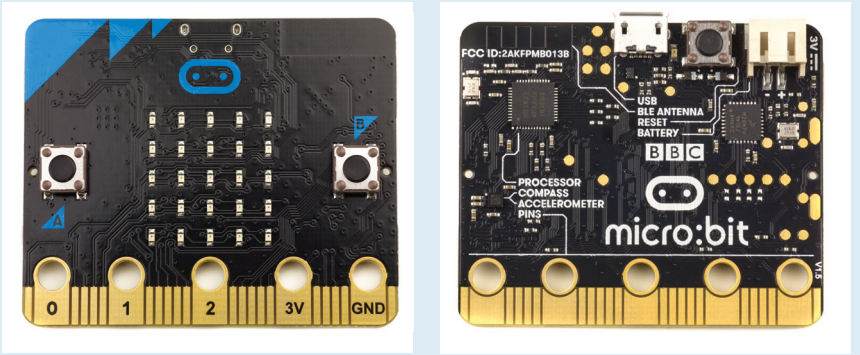
However, the BBC observed there was no prior technology on the market that suited the complete novice and that had been designed as an educational tool from the outset. For example, Arduino[19] set a new standard in the field, but requires wiring for virtually all of its projects as well as the installation of a custom IDE and device drivers. The Raspberry Pi is a highly capable device that runs a full operating system, but also has a reliance on additional peripherals to enable physical computing. Its associated high power consumption and complexity also means it cannot be easily run from battery power and embedded into children's projects. There also are cost implications for children, parents, and schools wanting to start making: devices and accessories need to be affordable enough to be accessible by children and parents from a variety of backgrounds.

Engaging, capable, hardware. Figure 1 shows (a) the front and (b) the back of the micro:bit, which measures 4cm x 5cm. Like many "development boards," the micro:bit is an exposed printed circuit board with all its components visible (in fact, explicitly labeled, as a learning opportunity). The micro:bit is designed to be engaging and interactive from the start: the front is designed to resemble a face with colored streaks of hair (upper left) and eyes as the logo (upper middle).

This playful design should not be mistaken for a lack of capability. The board is based around a modern 32-bit ARM Cortex-M processor (16kB RAM; 256kB non-volatile flash) and hosts an array of input/output capabilities including a 5x5 LED matrix,

PHOTO COURTESY OF PETER HELDEN, @PETERHELDENS.

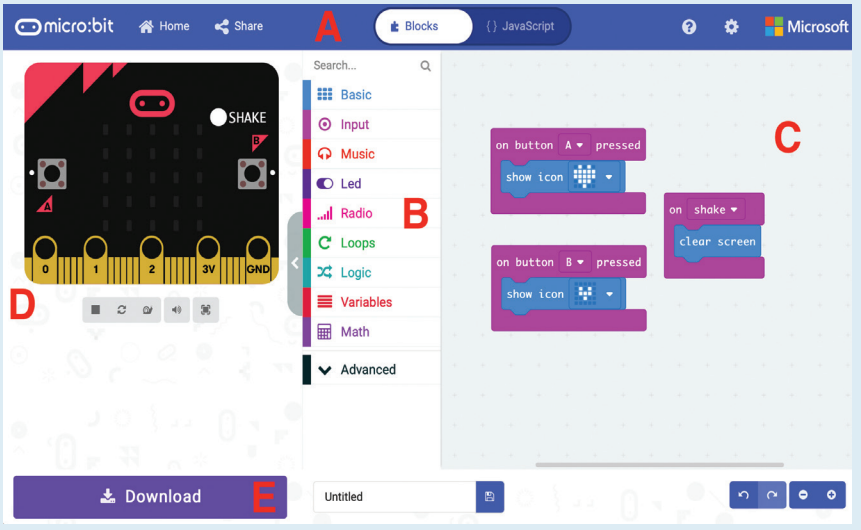
Figure 1. The BBC micro:bit.



(a) front, with two buttons, 5x5 LED display, and edge connector (bottom)

(b) back, with processor, accelerometer, compass, Bluetooth, USB and battery connectors

Figure 2. The MakeCode Web app for the micro:bit (https://makecode.microbit.org).



two programmable buttons, the ability to sense motion, gestures, magnetic fields, temperature and light. The device also includes a USB interface and edge connector with touch sensitive, digital/analog pins that allow external sensors, and actuators to be connected via crocodile clips or banana plugs. Finally, the device can communicate with phones, tablets, and computers via Bluetooth Low Energy (BLE) or directly with other micro:bits using a low-level 2.4GHz radio protocol. The ability to run on battery power and an ecosystem of micro:bit hardware peripherals that plug into the micro:bit's edge connector further expand its capabilities.

Engaging, simple, software. The design of the micro:bit coding tools is oriented toward a simple and inclusive starting experience with room for progression. In-school trials with a micro:bit prototype validated the BBC's approach of using a Web app based on the popular Blockly framework[8] for students to create scripts via the block-based visual programming paradigm pioneered by Scratch,[14] and providing a simulator for students to execute and debug their programs, all inside a Web browser.

In addition to block-based visual coding, support for text-based coding via scripting languages was identified as an important feature. As the micro:bit would be incorporated into standalone projects, it was essential for the user's program to be stored on the device for future untethered execution via battery power. This allows a student to unplug their micro:bit from a computer and show their creation to a teacher, parent or friend wherever or whenever they want.

The solution delivered by the BBC's partners includes support for Blockly, JavaScript and Python, all via Web apps. Figure 2 shows a screen snapshot of Microsoft's MakeCode (https://makecode.com) Web app for the micro:bit, which supports programming via both Blockly and JavaScript. The Web app has five main sections: (A) menu bar with access to projects/examples and switching between Blockly and JavaScript editors. To support progression, the editor also supports conversion of programs between Blocky and JavaScript—users can round-trip programs to see their code in visual or text-based representations; (B) Blockly toolbox of micro:bit API categories, representing the hardware capabilities of the micro:bit. This toolbox can be expanded through third-party extensions; (C) Blockly programming canvas showing a simple reactive program. MakeCode enables event-based programming through a lightweight scheduler in the underlying micro:bit runtime; (D) micro:bit simulator for execution of the user's program in browser; (E) download button, which invokes an in-browser compiler/linker to produce a binary executable (a "hex file").

The Python solution for the micro:bit is based on MicroPython (https://micropython.org), an implementation of Python 3.0 for microcontrollers. It includes a full Python compiler and runtime that executes on the micro:bit and supports a read-eval-print loop to execute commands sent via a terminal, for interactive use. This solution also allows a Python script to be embedded alongside the compiler/runtime and downloaded as a hex file from the Python Web app for the micro:bit (https://python.microbit.org).

A low-friction end-to-end experience. Figure 2C illustrates a simple coding example for the micro:bit, which displays a large heart when button A is pressed, 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). The interactive micro:bit simulator (Figure 2D) models all functions of the micro:bit and allows the user to test that the program works as expected. The shake event can be fired using a virtual button (white circle labeled "SHAKE"), or by moving the mouse back and forth rapidly over the simulator.

To generate a binary executable for the micro:bit, the user simply presses the "Download" button (Figure 2E), which invokes an in-browser compiler tool chain that translates the Blockly program to JavaScript and then to machine code, linking the user's compiled code against a pre-compiled C++ runtime.[6] This means that no C++ compiler is required for compiling the user's program into an executable binary; the same is true of the MicroPython solution.

When plugged into a host computer via USB, the micro:bit appears as a 'memory stick' storage device. A compiled program can be transferred (flashed) to the micro:bit by a simple file copy operation, installing the executable binary into the micro:bit's non-volatile flash memory. This makes it compatible out-of-the-box with almost all school computers and eliminates the complexity of installing device drivers—something that teachers and children rarely have permission to do. Once flashed, the micro:bit then can be embedded into projects where it runs on battery power.

Design summary. We conclude this section with a reflection on the five design goals stated earlier. (1) The micro:bit's inexpensive hardware lowers the financial barrier to entry for students, parents and teachers. Its Web-based software requires no installation, lowering technical barriers to adoption in schools and homes. The micro:bit's integrated sensors and outputs allow students to explore a range of lessons and projects without the need for external electronic components. (2) The design of the device prompts a sense of fun, alongside colorful programming blocks that allow for complete control over the device and its peripherals, backed up by a range of creative learning materials and projects. (3-4) Programming experiences spanning Blockly, JavaScript and Python provides a clear progression path when combined with project-based learning. Radio and Bluetooth networking allow further progression to more complex projects with other micro:bits, smartphones and other Internet connected devices. (5) Finally, the ability to run on battery power combined with sensors, non-volatile storage and edge connector allows for the integration of the micro:bit into areas of the curriculum that make use of physical experiments and data collection.

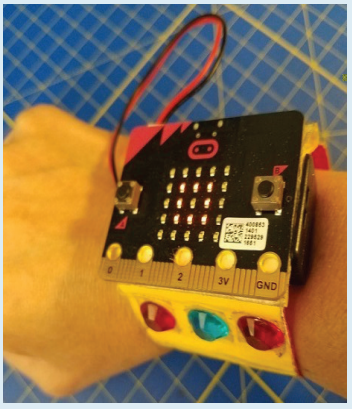
Projects

A wide array of curriculum-aligned lessons are available for the micro:bit. However, physical computing devices also lend themselves to creative (and often collaborative) projects that promote deep problem-based learning. Here we provide some examples of such educational projects for the micro:bit, grouped into four broad classifications of use, each showing many of our design goals in action.

Wearables and interactive play. Many projects involve the use of a micro:bit as an interactive mobile device—either as a handheld or wearable. Figure 3 shows a simple but highly popular micro:bit project: a micro:bit 'watch' that plays the rock/paper/scissors game by randomly displaying a rock (3x3 square), paper (5x5 square with center empty) or scissor icon on the 5x5 LED display when the device is shaken. Other popular examples include name and emoji badges, a graphical compass that points North based on magnetometer data, and gesture-based games such as 'Snake' which use the tilt of the device to control the behavior of objects shown on the LED display.

Digital crafting. Other popular projects augment a micro:bit with simple classroom supplies, allowing students to quickly create low cost, playful and practical digital artifacts. For example, Figure 4(a) shows how cardboard and aluminum foil can be used to build a competitive game known as 'Reaction.' Crocodile clips are used to connect pins P0, P1, P2, and GND of the micro:bit to conductive aluminum foil pads glued to a cardboard gameboard. Users are challenged to be the first to complete a circuit by touching the GND pad and one of other pads when the micro:bit display lights up. Note the blending of form and function evident in this design, including the positioning of the interface for multiuser

Figure 3. A micro:bit watch.



(a)
wearable form-factor rock/paper/scissors game

```
input.onGesture(Gesture.Shake, function () {
    let tool = Math.randomRange(0, 2)
    if (tool == 0) {
        basic.showLeds(`
            . . . . .
            . # # # .
            . # # # .
            . # # # .
            . . . . .
            `)
    } else if (tool == 1) {
        basic.showLeds(`
            # # # # #
            # . . . #
            # . . . #
            # . . . #
            # # # # #
            `)
    } else {
        basic.showLeds(`
            # # . . #
            # # . # .
            . . # . .
            # # . # .
            # # . . #
            `)
    }
})
```

(b)
the JavaScript of the game

access and the additional touch pad labeled "START."

Actuation adds a further dimension: the micro:bit can control servos and motors via its edge connector. This has resulted in the creation of inexpensive, cardboard based creations that can react to their environment, such as those shown in Figure 4(b): these simple robots open and close their mouths in response to light stimulus. Other similar examples include musical instruments, such as a 'guitar' that changes pitch based on its physical orientation, and goal line technology for tabletop football games.

Science and measurement. The micro:bit's small size and built-in sensors make it well suited to being embedded into science and technology projects for undertaking data measurement. A great example of this is provided by the Bloodhound project (http://www.bloodhoundssc.com), a U.K. initiative to set a new land speed world record. As part of their remit to inspire students about STEM subjects, the 'Race to the Line' project was launched across the U.K. In this project, students design, build, and race model rocket cars in competition, learning about physics, aerodynamics, engineering, and measurement. A micro:bit is integrated into the car's design, as shown in Figure 4(c). The micro:bit captures three-axis accelerometer data of the rocket car during its race. After the race, students upload the data from the micro:bit and analyze the performance of their cars.

Similarly, Figure 4(d) illustrates an environmental project that uses the micro:bit to measure soil moisture. The combination of water and nutrients in soil affect its conductivity—the more water, the greater the conductivity. This can be directly measured using metallic probes (note the use of inexpensive nails as probes here) and the micro:bit's integrated analog voltage sensor. Then, the micro:bit is programmed to periodically take a moisture reading and record the results into the device's internal flash file system for later analysis.

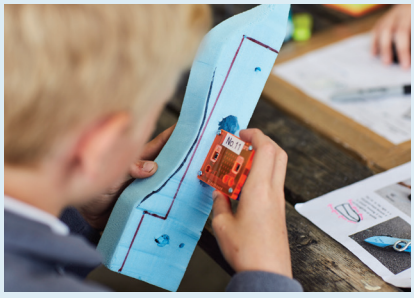
Interconnected devices. Our final class of projects are those that make use of multiple, wirelessly interconnected devices. The micro:bit has an inbuilt Bluetooth low energy (BLE) compatible 2.4GHz radio. BLE provides a private and secure mechanism through which the micro:bit can be programmed over-the-air from mobile phones and tablets, and also provides an API through which the micro:bit can be paired, and its sensors and actuators made available to applications running on such devices through a well-defined Bluetooth profile. MIT's Scratch 3.0 includes micro:bit support through this API, for example.

However, we observed the greatest level of innovation emerged from a simpler, custom-built packet radio protocol running on the same 2.4GHz hardware. With the micro:bit radio API, micro:bits can form low-level peer-to-peer multicast groups. Any data sent from one micro:bit is seen by all members of their group—thus

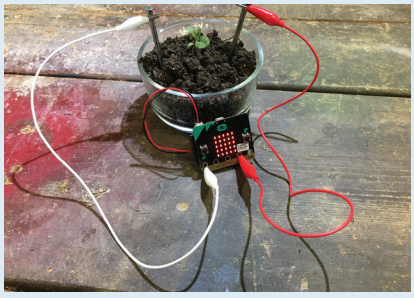
Figure 4. Example projects.

(a) the reaction game

(b) light-reactive cardboard robots



(c) a Bloodhound model rocket car instrumented with a micro:bit



(d) measuring soil moisture via micro:bit pins

Figure 5. A micro:bit-based vehicle controlled wirelessly by a second micro:bit.

enabling a simple yet powerful basis for projects involving group collaboration in a way not feasible with BLE. Examples here include remote control vehicles, such as that illustrated in Figure 5. This example uses two micro:bits sharing data over radio: one integrated into the vehicle to control steering and speed, and a second integrated into a handheld steering wheel that is used as a remote control. A second popular example is illustrated in Figure 6, which mimics how fireflies synchronize their blinking over time, as described in the accompanying Python program.

From the U.K. to the World

Here, we detail the history of the project, the approach taken to deliver 800,000 devices to students and their teachers in the U.K., and how the Micro:bit Educational Foundation is taking the micro:bit worldwide.

BBC micro:bit partnership. The BBC invited 29 partners to contribute hardware, software services, teaching materials, packing/distribution, logistics, events, and funding. These partners were not directly funded with public money by the BBC for their work on the project. Rather, partners contributed their own resources to make the micro:bit vision a reality.

The BBC looked for three types of partner for the original project: those who could help on the technical development; the manufacturing/distribution of the micro:bit, and, very importantly, those who could help with education—both child and the teacher. A large proportion of the 29 partners played a role in creating teaching resources, delivering teacher training and on-the-ground support in collaboration with grass roots organizations such as the U.K. Computing At School (CAS) network[5]—a group of over 30,000 computing teachers, supporters, and enthusiasts. These partners used such networks of practice to engage with over 90% of the 8,000 secondary schools in the U.K. even before the micro:bit was manufactured.

This activity was orchestrated and supported by a broad BBC team dedicated to raising awareness of the project. This essential activity ensured the teachers and students understood the aims and potential of the micro:bit ahead of the devices being available in schools. This team developed broadcast TV and media content with BBC talent like Peter Capaldi, will.i.am, and Paloma Faith, and major BBC brands including *Doctor Who*, *The Voice*, *Robot Wars*, and *Wolfblood*. They also organized nationwide public engagement activities including Make It Digital Roadshows that took place in 10 cities around the U.K. that had a combined footfall of approximately 100,000 people in the summer of 2015.

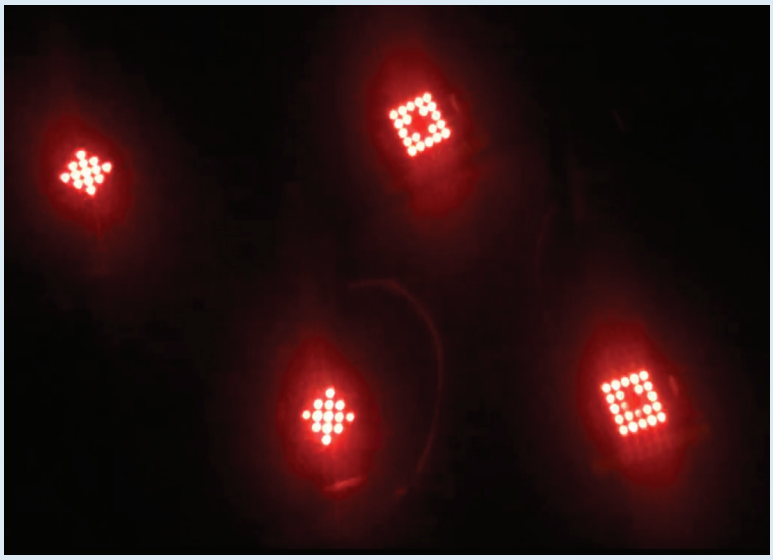
Once product manufacturing, testing, and certification was complete, 800,000 micro:bit devices were delivered into U.K. schools in March 2016—one device for every Year 7 (11/12 year old) child in the U.K. The micro:bit was well received by U.K. teachers and children, with high levels of engagement. In the first six months, there were approximately 13 million visits to the website, 10 million runs of the online micro:bit simulator, and two million programs downloaded to a micro:bit.

Expanding the scope. The U.K. micro:bit deployment sparked interest from around the world. In October 2016, the Micro:bit Educational Foundation was formed—a U.K.-based nonprofit organization that now serves as the custodian of the micro:bit legacy. Its vision is to inspire every child to create their best digital future, with focus on widening participation around gender and disadvantaged groups across the globe. It is funded through a small royalty on every micro:bit sold and by corporate sponsorship.

The foundation strives toward its vision by coordinating work across a broad partnership of educators, technologists, enthusiasts, and governments to bring about global change. Partnership has proven to be the vital heart of all the foundation's activities. More specifically, the foundation maintains partnerships with over 132 global organizations. These can be grouped into:

Hardware partnerships with manu-

Figure 6. Fireflies example: A visual representation of emergent behavior from a distributed algorithm, implemented in Python.



```
clock = 0
flash = []
for i in range(9, -1, -1)]:
    flash.append(Image().invert()*(i/9))

while True:
    incoming = radio.receive()
    while incoming == '0':
        clock = clock + 1
        incoming = radio.receive()
    if clock >= 8:
        radio.send('0')
        display.show(flash, delay=100, wait=False)
        sleep(200)
        clock = 0
    else:
        sleep(100)
        clock = clock + 1
```

micro:bit Artifacts

The BBC micro:bit hardware and software assets are open source at https://github.com, at the following locations:

- **/bbcmicrobit for the micro:bit hardware design, the micro:bit software prototype and theMicroPython port for the micro:bit;**
- **/microsoft/pxt-microbit for MakeCode for the micro:bit;**
- **/lancaster-university/microbit-dal for the micro:bit C++ runtime.**

A wide variety of educational resources for the micro:bit can be found at https://microbit.org. Resellers of the micro:bit and accessory manufacturers can be found at https://microbit.org/resellers.

facturers, suppliers, and resellers ensure a pipeline of micro:bits are available in 60 countries to date. Also essential are partnerships with accessory makers who create kits that enable projects such as those described earlier. Hundreds of third-party accessories have been created for the micro:bit. The most common examples include additional sensors (sound level, moisture, particulate, and ultrasonic ranging sensors), actuators (motor drivers, audio speakers, addressable LED, light strips), hardware prototyping kits that interface to breadboards, and finally, application-specific peripherals such as wheeled robots, remote controlled vehicles, and games.

Software partnerships with organizations including Lancaster University, Microsoft, and the Python Software Foundation ensure a diverse offering of programming languages and highly reliable, state-of-the-art editors for the micro:bit.

Countrywide partnerships with governments, charitable organizations, and regional companies enable trials and rollouts to schools around the world. To date, this has resulted in national scale deployment in Canada, Croatia, Denmark, Norway, Iceland, Singapore, Hong Kong, Uruguay, and the Western Balkan states, in addition to the U.K., with the British Council planning similar activity in the Western Balkan states in 2019.

Community partnerships that provide essential 'on the ground' support. Examples here include the generation and sharing of learning resources and experiences between teachers, crowdsourced translation of teaching materials into languages other than English, social media activists who reach out to hard-to-reach demographics, and volunteers that provide technical support.

Lessons Learned

We learned a number of lessons through the micro:bit project that might be helpful to others working in the space of CS education and physical computing:

Community-centered design. User-centered design was a key part of the BBC's approach to the micro:bit, considering a broad range of stakeholders including children, teachers, product developers, manufacturers, enthusiasts, and support organizations. This process identified physical computing as the main focus area and pointed the way to the key design decisions including the integrated board design (inspired by Arduino), and the need for block-based and scripting languages rather than the C-based sketches used by Arduino. Yet, as the project developed it became something greater—a process by which a community of practice emerged, consisting of those individual stakeholders. This enabled the strong and sustained ecosystem around the micro:bit, long after the initial U.K. project was completed.

Depth is just as important as ease of use. Although providing multiple layers of abstraction was central to realizing the "low-floor, high-ceiling" concept, we did not sacrifice the opportunity to dig deeper, learn key computing concepts underneath, and learn from the realities of the computing world. Take the packet-based broadcast radio interface, for example. This interface is entirely lossy, incorporating a simple checksum and providing no reliability guarantees. A user could send numbers using the radio to indicate states within a distributed application, but would need to include device identifiers as it scales, and if needed, algorithms for reliability. Before long the user has implemented their own networking protocol with real-world applicability.

An always connected experience is restricting. In the BBC prototype for the micro:bit, the text of a user's program was submitted to a compile service in the cloud that returned a final executable to be copied onto a micro:bit. We originally adopted this architecture for the micro:bit, but in trials across many schools in the U.K. found the assumption of "always connected" was not a good one. As a result, we eliminated the need for a cloud service by writing a compiler and linker in JavaScript that would produce the needed binary directly in the Web app. Once the Web app loads, no further connectivity is needed in order to edit, compile, and flash the program to the micro:bit.

Partnerships and localization are key to global expansion. The national scale deployments of micro:bit and further large-scale trials in countries such as Sweden, Taiwan, and Uruguay taught us that localization of all aspects of the approach is essential for success. Beyond the predictable challenges of language translation, there are many other aspects related to funding, educational priorities, and cultural diversity. For example: the U.K. rollout was funded entirely through donations from industry and charities; Iceland's rollout was funded by its government; and Croatia's rollout stemmed from a crowdsourced fund set up by a motivated regional entrepreneur. Likewise, teaching materials designed for the U.K. do not readily translate to other countries. Moreover, school projects are often based around local cultural events. We have learned that a combination of local and global partnership is the key to embracing diversity.

Compromises must always be made. To make the micro:bit hardware available to as many people as possible, it was critical to keep the cost low. This influenced the choice of components and capabilities of the hardware, which inevitably means trade-offs and limitations. For example, the

micro:bit's 16kB of RAM is quickly consumed, especially when the full Bluetooth stack is loaded, which can be frustrating for those who want to build larger applications. The 5x5 LED display is optimized to display a single ASCII character, but falls short when non-Latin character sets are considered, creating challenges for international adoption. The design of the micro:bit edge connector makes it easy to plug the micro:bit into another board, but is inherently non-compositional, compared to the approach of Arduino that allows stacking of boards via its headers. Without these difficult choices, the micro:bit would not have become a reality.

Timing is critical. As with any complex endeavor, luck favors the prepared. The U.K. was the first country to mandate computing education for K–12, there was a large group of volunteer computing organizations in the U.K. to call upon, and Moore's Law had brought microcontroller and networking technology (such as Bluetooth) down in cost so as to enable delivery at scale economically.

Outcomes

Since the initial distribution of micro:bits in 2016 we have observed significant interest, enthusiasm, and adoption. The BBC micro:bit is now available in 60 countries and 24 languages, and in excess of four million devices have been delivered to end users globally with an increasing demand year over year. The online editors have hundreds of thousands of independent sessions every month. Activity on social media and support networks also indicate high levels of use for micro:bit resources in schools, particularly those related to the constructionist pedagogy.

Independent research undertaken in the U.K. (see https://microbit.org/ta/2017-07-07-bbc-stats by Discovery Research) supports these observations.[17,18] An independent survey[2] of 405 U.K. school children and their teachers concluded that:

- 86% of students said the micro:bit made computer science more interesting;
- 70% more girls said they would choose computing as a school subject after using the micro:bit;
- 85% of teachers agreed it made ICT/computer science more enjoyable for their students;
- Half of teachers who have used the micro:bit said they felt more confident as a teacher, particularly those who said they were not very confident in teaching computing.

Although preliminary studies are encouraging and guide our thinking, they are small compared to the scale of the BBC micro:bit project. It will take more time to determine the full impact of the micro:bit. We look forward to further research studies that will investigate the advantages and challenges of using the micro:bit in supporting teaching and learning, both within computing and in wider cross curricular ways.

Acknowledgments

The BBC micro:bit project's 29 partners were Arm, Barclays, Bluetooth Special Interest Group, Cannybots, Creative Digital Solutions, Cisco, Code Club, Code Kingdoms, CoderDojo, CultureTECH, element14, the Institute of Engineering and Technology, Kitronik, Lancaster University, London Connected Learning Centre, Microsoft, MyMiniFactory, National STEM Centre, Nordic Semiconductor, NXP, the Python Software Foundation (PSF), Samsung, ScienceScope, STEMNET, Tangent Design, Technology Will Save Us, Teen Tech, the Tinder Foundation, and the Welcome Trust.

The authors also recognize the contributions of Jeannette Wing, Zach Shelby, the Computing At School organization and its members, Damien George and Nicholas Tollervey at the PSF, Clare Riley, Jonathan Protzenko, Michael Braun and the MakeCode team at Microsoft, Martin Wooley at the Bluetooth SIG, Mike Powell and Jonathan Smith at Premier Farnell, Jo Claessens, Cerys Griffiths and the many other colleagues across the BBC who supported this initiative. Without the tireless contributions from all these organizations and individuals the micro:bit project would not have been a success.

References

1. BBC. The BBC micro:bit, 2015; http://www.bbc.co.uk/programmes/articles/4hVG2Br1W1LKCmw8nSm9WnQ/the-bbc-micro-bit.
2. BBC. BBC micro:bit celebrates huge impact in first year, with 90% of students saying it helped show that anyone can code, 2017; https://www.bbc.co.uk/mediacentre/latestnews/2017/microbit-first-year.
3. Blikstein, P. Digital fabrication and 'making' in education: The democratization of invention. *FabLabs: of Machines, Makers and Inventors*, 2013, 4:1-4:21.
4. Blyth, T. The legacy of the BBC micro: Effecting change in the U.K.'s cultures of computing. May 2012; https://media.nesta.org.uk/documents/the_legacy_of_bbc_micro.pdf.
5. Crick, T. and Sentance, S. Computing at school: Stimulating computing education in the U.K. In *Proceedings of the 11th Koli Calling Intern. Conf. Computing Education Research*. ACM, 2011, 122–123.
6. Devine, J., Finney, J., de Halleux, P., Moskal, M., Ball, T., and Hodges, S. Makecode and CODAL: Intuitive and efficient embedded systems programming for education. In *Proceedings of the 19th ACM SIGPLAN/SIGBED Intern. Conf. on Languages, Compilers, and Tools for Embedded Systems*. LCTES 2018, 19–30.
7. Dougherty, D. The maker movement. *Innovations: Technology, Governance, Globalization 7*, 3 (2012), 11–14.
8. Fraser, N. Ten things we've learned from Blockly. In *Proceedings of the 2015 IEEE Blocks and Beyond Workshop*, 49–50.
9. Halverson, E.R. and Sheridan, K. The maker movement in education. *Harvard Educational Review 84*, 4 (2014), 495–504.
10. Hodges, S., Scott, J., Sentance, S., Miller, C., Villar, N., Schwiderski-Grosche, S., Hammil, K., and Johnston, S. .NET.Gadgeteer: A new platform for K-12 computer science education. In *Proceeding of the 44th ACM Technical Symp. Computer Science Education*, ACM, New York, NY, USA, 2013, 391–396.
11. Papert, S. *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books, 1993.
12. Peppler, K. STEAM-powered computing education: Using E-textiles to integrate the arts and STEM. *IEEE Computer*, (2013), 1.
13. Peyton Jones, S. Computer science as a school subject. In *Proceedings of the 18th ACM SIGPLAN Intern. Conf. Functional Programming*. ACM, 2013, 159–160.
14. Resnick, M. et al. Scratch: Programming for all. *Commun. ACM 52*, 11 (Nov. 2009), 60–67.
15. Resnick, M. and Silverman, B. Some reflections on designing construction kits for kids. In *Proceedings of the 2005 Conf. Interaction Design and Children*. ACM, 2005, 117–122.
16. Robelen, E.W. STEAM: Experts make case for adding arts to STEM. *Education Week 31*, 13 (2011), 8.
17. Sentance, S., Waite, J., Hodges, S., MacLeod, E., and Yeomans, L. 'Creating cool stuff:' Pupils' experience of the BBC micro:bit. In *Proceedings of the 2017 ACM SIGCSE Tech. Symp. Computer Science Education*. ACM, 2017, 531–536.
18. Sentance, S., Waite, J., Yeomans, L., and MacLeod, E. Teaching with physical computing devices: The BBC micro:bit initiative. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*. ACM, 2017, 87–96.
19. Severance, C.R. Massimo Banzi: Building Arduino. *IEEE Computer 47*, 1 (2014), 11–12.

Jonny Austin is chief technology officer at the Micro:bit Educational Foundation, London, U.K.

Howard Baker is an education researcher at the Micro:bit Educational Foundation, London, U.K.

Thomas Ball is a partner researcher at Microsoft Research, Redmond, WA, USA.

James Devine is a Ph.D. student in the School of Computing and Communications at Lancaster University, U.K.

Joe Finney is a professor in the School of Computing and Communications at Lancaster University, U.K.

Peli de Halleux is a principal research software development engineer at Microsoft Research, Redmond, WA, USA.

Steve Hodges is a senior principal researcher at Microsoft Research, Cambridge, U.K.

Michał Moskal is a principal research software development engineer at Microsoft Research, Redmond, WA, USA.

Gareth Stockdale is chief executive officer at the Micro:bit Educational Foundation, London, U.K.