James **Devine**

A strong, user-focussed systems researcher and tool builder with a history of impactful research and innovation.

james_a_devine@outlook.com | jamesadevine.com | jamesadevine | James Devine

Rust · Python · TypeScript · C · C++ · Bash · Powershell · YAML (Pipelines)

Education

Lancaster University
Lancaster, England

Ph.D. Computer Science — *Jan. 2016 - May. 2020*

Undertook research on lowering the barrier to entry for innovation with microcontroller-based devices. Collaborated with companies, including Microsoft, ARM, Farnell, and Samsung to deliver the **BBC micro:bit**. Contributed the device runtime for the micro:bit used by millions of children around the world via **Microsoft MakeCode**. Generalised the runtime to support a range of other products, allowing Microsoft MakeCode to diversify into different domains. Applied research to enable private IoT in educational settings for educators and students to reduce energy consumption. Designed and built a custom protocol called **Jacdac** to support plug-and-play physical computing. The protocol was used to help fashion designers integrate and program technological garment for fashion show in Brooklyn. Jacdac is integrated in shipping products from Microsoft, Elecfreaks, Kittenbot, and Forward Education.

Lancaster University
Lancaster, England

B.Sc. Computer Science, First Class (Hons) — *Oct. 2012 - June 2015*

In my 3 year undergraduate program, I started my own 3D printing design company, led a prizewinning group project, and spent two years developing the university's flagship smart phone application whilst achieving a grade within the 95th percentile of my cohort. For my dissertation project I created a per-appliance current sensing device, accompanying IoT infrastructure, and a cross-platform smart phone application for visualising data.

Experience

Technical Lead, Microsoft
Cambridge, UK

Azure Sphere — *February. 2025 - Present*

Azure Sphere is an end-to-end hardware security platform for IoT that allows customers to build products without worrying about security. Azure Sphere is used in critical customer applications and has hundreds of thousands of monthly active devices. I lead the team responsible for maintaining the OS and accompanying customer-facing tooling.

- Maintained 100% compliance rate against customer security SLAs
- Led an AI-first engineering culture within the team to improve productivity; leading to an average 1.17X improvement in productivity.
- Defined and assisted the team in delivering 95% of the work items for the current period ahead of schedule.
- Delivered high-profile updates to engineering leadership on team progress and org wide updates on engineering best practice for AI tooling.

Senior Software Engineer, Microsoft
Cambridge, UK

Azure Sphere — *April. 2022 - February. 2025*

My initial role with **Azure Sphere** was customer acceleration. Responsibilities included handling customer support escalations, build tooling to ease the application development experience, and writing samples to demonstrate best practice for customers to follow. Due to my strong software engineering skills, after a year, I was moved to lead a team focussed on improving the quality and security of the entire product while continuing to champion customer acceleration.

- Defined and implemented work that improved project pipeline success rate from 49% to over 85%.
- Resolved over 25 high impact customer support escalations.
- Single-handedly wrote and shipped a systemd daemon in Rust to improve the Linux user experience.

Researcher, Microsoft Research
Cambridge, UK

Democratizing Hardware — *June. 2020 - April. 2022*

Drove research on tools, systems, and experiences that make it easier to produce prototypes (one-off) and convert them to isotypes (many-off) hardware at scale. Fundamental to this research is **Jacdac**, a technology contributed in my PhD thesis.

- Led **Project MakeAccessible**, a hackathon project centred around empowering more people to build custom assistive technology. Jacdac hardware kits were shipped to hackers, and over 80 hackers participated across 4 continents.
- Designed and developed the micro:bit **MakeCode Arcade shield.** manufactured and shipped by over six companies
- Designed and developed software, firmware, and hardware for over 20 Jacdac modules manufactured and shipped by three companies.

Intern, Microsoft Research

Redmond, USA

Project Brookdale

Feb. 2019 - May 2019

Used Jacdac and MakeCode to empower fashion designers to develop intuitive wearable fashion technology. The platform was deployed and evaluated at a high profile fashion show in Brooklyn, New York. Fashion designers were able to realise their design vision for fashion-tech garments by embedding microcontrollers and sensors and dynamically integrating them via Jacdac. This was documented in a **Microsoft Research blog post.**

- Worked with 15 fashion designers and models in a high-stakes environment at a Brooklyn Fashion Show.
- Designed and manufactured purpose built Jacdac devices used in the fashion show.
- Developed the Jacdac TypeScript stack to enable programs to be written in Microsoft MakeCode.
- Presented Jacdac and Project Brookdale the platform at high profile conferences.

Intern, Microsoft Research

Redmond, USA

Jacdac

Jun. 2018 - Sep. 2018

Created a wired networking protocol for dynamically integrating embedded devices and peripherals. Think USB, but more flexible and designed for ultra low-cost microcontrollers. Jacdac is used as the interconnectivity solution for **MakeCode Arcade** devices.

- Defined and developed the Jacdac protocol stack from the physical layer, to the control layer, to the software driver models used by developers.
- Implemented the protocol on three different processor classes to prove viability, including a 4-cent OTP microcontroller.
- High profile papers at top tier conferences: PLDI, Ubicomp (Best paper)

Intern, Microsoft Research

Redmond, USA

Embedded Learning Library (ELL)

Jun. 2017 - Sep. 2017

Undertook an internship with the **ELL** team in Microsoft Research Redmond. ELL is an LLVM backend for Pytorch targetting microcontrolller class applications. I created a wake-word recognition solution (like "Hey Cortana!") for resource-constrained microcontrollers, including developing the fundamental neural network models in LLVM and making them accessible from Pytorch.

- Learnt AI fundamentals such as the theory of recurrent neural networks and their role in machine learning.
- Developed a memory and processor efficient C implementation of mel-frequency cepstrum cepstral coeffient calculations for microcontroller class devices.
- Implemented neural networks in LLVM and C++: LSTM, GRU, RNN from first principals for cross-compiling to microcontroller targets
- Added binarisation for ELL to reduce on-device model size

Research Associate, Lancaster University

Lancaster, England

The BBC micro:bit, MakeCode, and CODAL

Jun. 2015 - 2020

Co-wrote the micro:bit runtime, a memory efficient lightweight operating system designed to support higher level languages like JavaScript. Later generalised the micro:bit runtime into CODAL, which now supports upwards of 50 devices in the MakeCode programming editor. **The BBC micro:bit** is a small embedded physical computing device that was given to 750,000 11–12 year old students in the UK in 2015. Designed to provide an engaging, low barrier way to learn computer science concepts, there are now over 10 million micro:bits in use worldwide.

- Design and develop the micro:bit runtime, a lightweight and efficient operating system that runs in less than 2 kB of RAM.
- Delivered the micro:bit now used by millions of children all around the world.
- Collaborated with partners, including ARM, Farnell, Samsung, the BBC, and Microsoft.
- Continued involvement in design discussions and future directions for the **micro:bit foundation**, and Microsoft MakeCode. I also add new devices to the MakeCode ecosystem using CODAL

Publications

Rethinking the Runway: Using Avant-Garde Fashion To Design a System for Wearables
Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021

The BBC micro: bit–from the UK to the World
Communications of the ACM (2020). ACM, 2020

Enabling intuitive and efficient physical computing
Thesis (2020). Lancaster University, 2020

MakeCode and CODAL: Intuitive and Efficient Embedded Systems Programming for Education
Proceedings of the 19th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems, 2018

Jacdac: Service-Based Prototyping of Embedded Systems
Proc. ACM Program. Lang. 8.PLDI (June 2024). Association for Computing Machinery, 2024

MakeDevice: Evolving Devices Beyond the Prototype with Jacdac
Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction, 2023, Warsaw, Poland

Plug-and-play Physical Computing with Jacdac
Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 6.3 (Sept. 2022). Association for Computing Machinery, 2022

Understanding How People with Limited Mobility Use Multi-Modal Input
Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, New Orleans, LA, USA

Full publication history available on Google Scholar.