A micro:bit style Edge AI platform

William Dove
School of Computing & Communications
Lancaster University
Lancaster, UK
w.dove@lancaster.ac.uk

The BBC micro:bit has become a core educational tool, with usage both as a tool for education at a variety of levels functioning not only as a tool to teach programming at school, but also allowing embedded systems concepts to be explored at a university level and also function as a data collection platform for subjects such as biology. While the software and hardware platform are highly capable, higher performance and more robust systems are necessary to make it suited for field use for more serious projects. To achieve this we build an improved platform featuring higher performance hardware, and enhanced networking to allow integration into robotics and sensor networks.

I. Introduction / Background

The BBC micro:bit is a compact, compute kit designed for education applications. It was originally designed for usage by secondary school age children to teach them computer science concepts by providing an intuitive and exciting platform which allows them to interact with the physical world via a variety of sensors and other devices by a range of methods such as radio and wired communication.

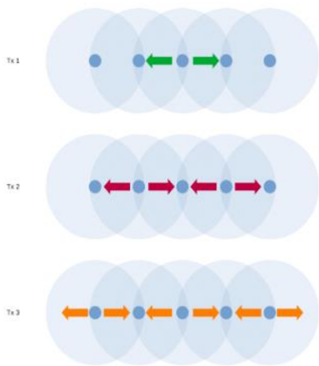
It has been increasingly used in more “production” like settings where it is being used for its characteristics as a cheap compact single board computer which can easily be expanded on for deployment in a range of environments. This can involve simple classroom projects measuring soil water content for biology studies up to more advanced data collection in the field.

It has also started to be utilised in simple AI settings. For instance a model has been created to detect body movements with the accelerometer so movements can be used as programming event.

Therefore I propose a device based on the micro:bit, taking advantage of recent hardware and software improvements to provide a platform for sensors to be built on. This will feature a high performance quad core nRF54H20 and edge TPU unit to allow high performance which when combined with a mesh network can make a platform suited for robotics and sensor networks, including those with actuators linked within the local network as the AI can be used to tune their behaviour, for instance this could be used for a water quality monitoring network when attached to actuators which can control the level to maintain a legal pollution level.

II. Related work

For localised networking I propose the usage of a concurrent flooding protocol[2]. This has the advantage of implicitly minimising the effect of collisions during the retransmission of packets, as the interference is minimal when the packets are transmitted simultaneously(within one to 2 cycles of a 16MHz Clock). However, the main disadvantages of this method is that it can only work over reasonably short distances as the physical behaviour of how it works at longer ranges has yet to be evaluated. As this is a flooding protocol, it works best for situations where data is being transmitted from a central device such as some sort of controller to all devices in the network, as during point to point transmission its likely most devices receiving a message will not require it, however this waste can be reduced via the usage of directed flooding. The figure below shows the concurrent retransmission system.



Over longer distances, a more conventional routing based network will likely be necessary such as AODV[3], or if longer distance hops are less likely it may make sense to transfer messages to an internet capable device such as a raspberry pi so they can be forwarded over any distances via either wired or wireless connections.

Edge machine learning[4] is an emerging field with the aim of placing inference on the edge. This has many advantages as it means data does not need to be transmitted to the cloud this obviously improves data security, and when network issues arise, this also makes the system robust. There is also the potential to combine edge machine learning in the cloud, by using edge machine learning to preprocess data to reduce the volume of data that must be transmitted. Use cases include optimizing the performance of wireless networks by optimising routing or scheduling. While it is possible to run inference on CPUs for simple tasks, for more advanced tasks such as those involving the processing of image data it tends to be run on dedicated hardware such as TPUs which are designed to parallelize the calculations.

III. Imagined or existing prototype sketches/drawings/photos

The hypothetical device will be based on a form factor similar to the BBC micro:bit to enable it to be compatible with some existing accessories. It will be extended with the presence of an integrated TPU to provide enhanced ML capabilities on the edge. To facilitate this the processor will have to be swapped from an nRF52832 to a nRF54H20 as this processor has USB2.0 capabilities which allow it to interface with the TPU, the TPU is described below.

A. nRF54H20

The nRF54H20 is Nordics highest performance processor, designed to provide a high performance embedded platform. It features 4 user accessible cores. These are 2 ARM cortex-M33 cores, which are designed to be used for the application core and radio core. There are also 2 RISC-V cores, one is a high performance core designed to be used as a software defined peripheral, such as running inference on, so this does not interfere with process happening on the main core, the other RISC-V core is a low powered core designed for peripheral control, such as potentially being woken on an input and running validation before turning other cores on for more complex processing.

B. Coral Accelerator Module

The coral Accelerator module is a small chip which includes Googles edge TPU. This offers 4 TOPS at a power of 2W. Making it more power consuming than the nRF54H20, while also providing significantly higher performance for ML tasks. It can be interfaced with via both PCIe and USB2.0. for our uses the USB interface will be taken advantage of as the nRF54H2) does not have the ability to interface with PCIe devices. This supports more complex models such as those written in TensorFlow for image classification.

C. Arducam Mini

The arducam mini is a 2MP camera which can be interfaces with via either SPI or I2C.This means it can be accessed from the nRF54H20. This will provide image data which can then be processed.

D. Prototype

During the prototyping process, the first iteration of the device will involve combining a nRF54H20 development kit with the other components as this allows the software integration to be tested to ensure it is feasible, before the development of a custom board.

As the project continues the hardware will be integrated onto a micro:bit sized platform, while the camera or any other sensors used to provide input for processing will stay externally mounted to allow them to be dynamically positioned to make them compatible with a range of use cases, these may be connected via a protocol such as JACDAC[1].

IV. Responsible Innovation

The prototype only takes advantage of a small number of components and when networked can interface with other off the shelf devices to minimise the costs of building the system out, this also allows existing devices to be used so they can be reused after testing.

Also by moving machine learning capabilities closer to the edge it reduces the energy consumption as large amounts of data do not need to be processed centrally on the cloud.

V. Author Bio(s) / Experiences

My primary experience with firmware development comes from my time developing a concurrent flooding protocol for the BBC micro:bit, under the supervision of Joe Finney.

I also have experience with AI, primarily from my 4th Year HCI project where my group developed a system to record and classify the sound of a user clicking their teeth to allow this to function as an accessible input device.

I also have AI experience from my 4th Year dissertation which takes advantage of LLMs to generate node-red flows to interact with Web of Things Devices. This makes the automation of IoT processes both more efficient and substantially quicker, being a useful tool to allow users to experiment with how they want to connect devices.

I also have performed well in modules studying networking, having developed an understanding of a range of networking protocols.

VI. Acknowledgements

I’d like to thank Joe Finney for his guidance and supervision with my third year project, which is the major reason behind my practical experience with firmware development.

VII. References

[1] J. Devine et al, “Plug-and-play Physcial Computing with Jacdac”, in Proceedings of the ACM n Interactive, Mobile, Wearable and Ubiquitous Technologies.vol 6, 2022

[2] W. Dove, Coe: Concurrent Flooding on the BBC Micro:bit, unpublished, 2024

[3] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, 1999, pp. 90-100, doi: 10.1109/MCSA.1999.749281.

[4] W . Li. “A Comprehensive Review and a Taxonomy of Edge Machone Learining: Requirements, Paradigms, and Techniques.”, MDPI AG, 2023