Introduction to the BBC micro:bit

Programming for beginners with the BBC micro:bit

John Davies

2026 May 13

1 Introduction

Synopsis

Wikipedia describes the micro:bit as a 'single-board microcontroller'. The 'single [printed circuit] board' part is obvious but what is a **microcontroller**?

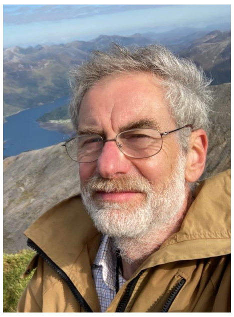
By 'computer' you probably think of a desktop computer, laptop or smartphone. But your house and other possessions are full of small computers called **microcontrollers** in **embedded systems** that you probably don't realise are there.

In this talk I'll cover:

- who am I
- brief history of microprocessors, microcontrollers and embedded systems
- examples of embedded systems in daily life
- main components of a small microcontroller (computer in embedded system)
- ARM, a major provider of designs for processors, based in Cambridge
- the BBC micro:bit

About me

- Worked in James Watt School of Engineering at University of Glasgow, 1986–2020.
  - Most research was on tiny semiconductor devices
  - Taught microcontrollers and embedded systems from 2000: lectures, labs, team projects, individual projects.
  - Led academic aspects of transnational education in Singapore and Chengdu, China.
  - Dean of Learning and Teaching for Science and Engineering, 2014–2019.



- Accreditor for Institution of Engineering and Technology (IET), 2007–2022.
- Engineering Professors' Council, 2011–2017.
- Enjoy nature conservation, cycling and hillwalking.
- Currently membership secretary of Bearsden and Milngavie u3a.

2 History of microcontrollers

The *Atlas* family – Chilton upper floor (1964–1973)

Punched cards, paper tape, lineprinter, magnetic tapes, ... Image from Atlas Computer Laboratory at Chilton.

For people of my generation (mid-sixties), TV clips of computers almost always focussed on tape drives, manically driving forwards and backwards. Magnetic tape was the main medium for transferring data between systems, a function later provided by floppy disks, CDs and thumb drives. Tape was also important because computers themselves had only small memories, so intermediate results often had to be stored temporarily on tape – again like floppy disks on early personal computers. You can see the drives in the background of this photo of the *Atlas* computer at Chilton, with operators loading or unloading two drives. The two people in the front are working with punched cards and paper tape, while the woman behind them is watching her output emerge from the lineprinter. All of these activities are familiar to people of my generation who used computers.

The Atlas was an early supercomputer and different models ran from 1962 to 1976. Wikipedia has information and links at Atlas_(computer). Historical information is available at Atlas Computer Laboratory, Chilton: 1961–1975 and Simon Lavington wrote an extensive report on this computer. A friend of mine (Alan Clarke) worked on the longest-surviving Atlas at the CAD Centre in Cambridge, which ran until late 1976.

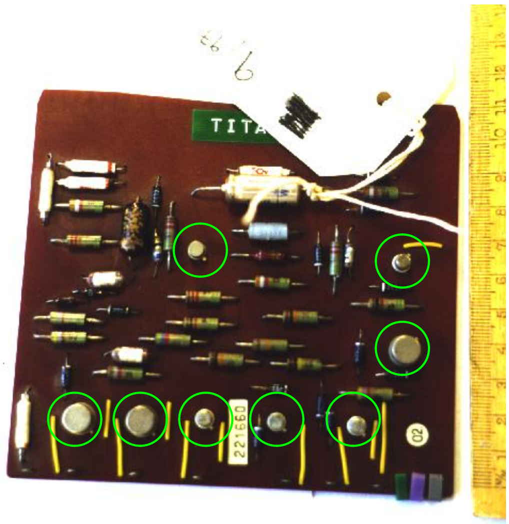
This is a model of the overall system, closely resembling that at Chilton (in History of Science collection, National Museums Scotland). The computer itself is on the lower floor, including the central processing unit (CPU) and memories. It is easy to imagine that the design of the CPU was the principal challenge in early computers but in fact it was the high-speed memory used by the CPU. Atlas had an advanced memory system with two physical types of memory to expand the apparent memory available to the CPU (virtual memory). The lower floor is roughly equivalent to the motherboard of a personal computer. Power supplies were probably downstairs as well. The input and output devices are on the upper floor including magnetic tape drives, paper tape readers, punched card readers and printers (lineprinters and teletypes). This is something like the user interface of a desktop computer. The operators' desks are in the middle; I suspect that there should be only one (and that it should be on the lower floor) but the model was made in two halves and each has one.

The photograph shows only the hardware but another important feature of Atlas was its supervisor programme, which we would now call the operating system. I suspect that both this software and the memory system ensured its long life in service.

There is no sign of networking and this was long before the internet but computers were often linked, even in the early days, and this includes Atlas in its later years. Programs were written using punched paper tape or cards in the early days but teletypes were connected to the computer to improve the productivity of programmers as time went on; Titan and the later Atlas machines were redesigned with this in mind.

The National Museum of Scotland (NMS) has a significant collection of the hardware from the Chilton Atlas, acquired in 1973 through the foresight of the Keeper of the Department of Technology at the Royal Scottish Museum, Dr A G Thomson. He commented that 'the rapid rate of technology obsolescence in the computer field has meant that there is a real danger that many important machines may not be preserved within adequately safeguarded collections, and in particular that only small portions of larger machines may survive'.

The *Atlas* family – Titan circuit board (1964–1973)



From Relic Information, Computer Laboratory, Department of Computer Science and Technology, University of Cambridge.

One member of the Atlas family was developed in collaboration with Cambridge University and was called *Titan*. It ran from around 1965 to 1973. I went up to Cambridge as a student in 1974, too late to use it myself, but I met several people who had used Titan. One person had the output from his final job on Titan proudly displayed on the wall. Wikipedia has information and links at Titan_(1963_computer).

The electronics was built from discrete (individual) transistors – this was before the days of integrated circuits. In fact it was before the days of silicon because the transistors, such as the OC170, were made of germanium instead. The transistors are in the cylindrical metal cans with tabs; the other components are mostly resistors (long cylinders with coloured bands), diodes and capacitors. Transistors act as switches in a digital system like a computer; early digital computers used electro-mechanical switches such as relays.

Note the scale. This board is larger than a typical smartphone but contains only 8 transistors and performed only a simple function, perhaps just a single logic gate such as NAND. A current smartphone contains more than 8 *billion* transistors in its processor alone (the memories contain far more) and is vastly more powerful than Atlas; it also uses a tiny fraction of the power, which is perhaps even more impressive. (This board may have been added to the computer locally, rather than part of the commercial product, because it has a rather hand-made appearance.)

Many people are astonished to learn how many transistors are in their smartphones. The approximate figure of 8 billion refers to the processor alone and some processors in phones contain more: the Apple A16 Bionic, launched in 2022 in the iPhone 14 Pro and 14 Pro Max, contains 16 billion transistors (Wikipedia page on Apple A16). The microprocessors in a laptop or desktop computer contain even more transistors; the Apple M1 Ultra has 114 billion for example, a number that I find astonishing. The memories contain even more – roughly one transistor per bit of memory (8 transistors per byte), although this depends on the specific technology used. A modern phone might contain 512 gigabytes of flash memory, which alone means 4 trillion bits (4×10^{12}) and possibly that number of transistors.

Only six Atlas systems were made. Each cost over a million pounds at the time (in pre-decimal currency!). Smaller systems called *minicomputers* also became popular and led ultimately to *microprocessors*, on which almost all current computers are based.

A well-known example of a minicomputer was the PDP-8, made by Digital Equipment Corporation (DEC). The photo shows a display at the National Museum of Computing, Bletchley, England. Wikipedia has information and links at PDP-8. It was made from about 1965–1974, only a little later than Atlas. The photograph shows the equivalent of the motherboard of a modern personal computer, including the CPU and high-speed memory. Early models cost $18,000 but the price fell below $10,000 as individual transistors were replaced by integrated circuits, which reduced the size and complexity of the assembly. Later models could fit all the electronics inside the base unit and dispensed with the cabinet above. A complete installation would include further, slower storage and input–output devices such as printers.

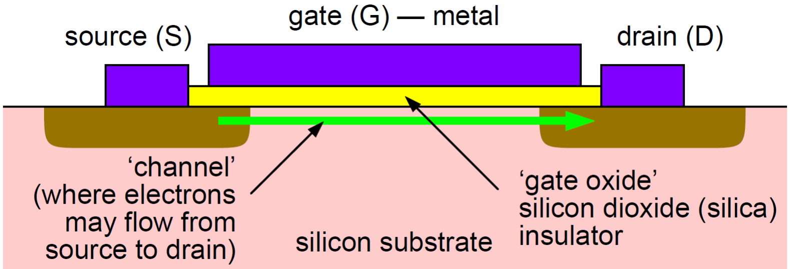
The PDP-8 operated on words of 12 bits, which you can deduce from the number of bulbs and switches in each row of the console. Modern computers almost always operate on multiples of 8 bits but this was not standardised until around 1980.

Later models followed the PDP-8. Another well-known family was the PDP-11, which operated on words of 16 bits. It was in turn developed into the 32-bit VAX-11 and I did many of the calculations for my PhD on a VAX-11/780 in 1980–81.

Minicomputers were adopted in factories and laboratories, where they could work alongside equipment and perform the sort of task for which computers are now taken for granted. Modern personal computers or PCs (originally called microcomputers) are the descendants of the PDP-8 and similar machines. They extended the usage beyond the likes of Atlas, whose job was large-scale computation in isolation.

What are transistors and integrated circuits?

Digital circuits work in binary – 0s and 1s – and operate by opening and closing switches. In silicon the switches are a type of transistor called a MOSFET. Here is a wildly over-simplified sketch of an antique device.



The voltage on the **gate** opens and closes the **channel**, through which current flows between the **source** and **drain**.

MOSFETs come in two complementary 'polarities', hence the **C** in **CMOS**. Many transistors are connected on the silicon to form an **integrated circuit** (**IC** or **chip**).

A modern MOSFET barely resembles this sketch but the principle is the same.

Developments over the decades

A smartphone out-performs any of the computers that I have shown. The advances are driven by astonishing developments in integrated circuits (ICs or chips). A large IC contains *billions* of transistors, whose length is now below 10 nm (10×10^{-9} m): around 100 atoms long. Smaller transistors are faster and use less power.

- By 1971 an integrated circuit could hold the complete central processing unit (CPU) of a computer: the **microprocessor**. This led to microcomputers, including the BBC Micro.

  Microprocessors increased speed, power and complexity: multiple cores. . . .

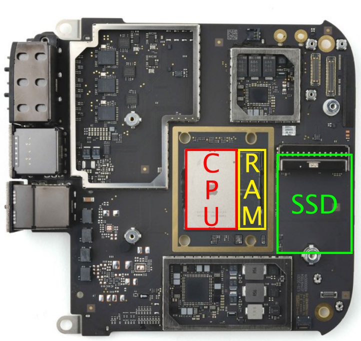
  Current examples in personal computers: Intel Core, AMD Ryzen, Apple M4.

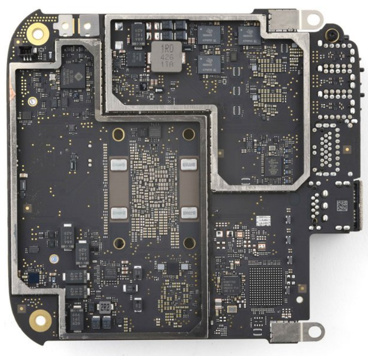
- A few years later, memory and other peripherals were included with the CPU to produce a complete **computer on a chip**: the **microcontroller** (MCU).

  The range has grown with both high-performance and small, cheap devices.

Most digital processors are in microcontrollers rather than microprocessors.

Inside a Mac mini





Top and bottom of the main circuit board, about 110 mm square.

Other components: communications, specialised memories, clocks, power. . .

From iFixit's Repair Guide to the Mac mini (2024).

Apple also publish a repair manual with a good exploded diagram.

- I have labelled the applications processor as CPU but in fact it contains at least 36 processors of different types.

- The RAM is the fast memory used for programs and their data while they are running; the contents are lost when the power is turned off.

- The SSD is the solid state drive, the modern replacement for a hard disk drive. This holds programs and documents more-or-less permanently. The module was removed at an earlier stage of the teardown.

- The silver boxes on the far left cover the communication ports on the back of the computer: USB, HDMI and ethernet. The space is occupied by the mains socket but the power supply is on another board.

As we have seen, Atlas was built from individual (discrete) transistors, each of which was packaged in a small metal can. However, the technology was soon developed to fabricate more than one transistor on silicon and wire them together into a circuit. The whole circuit was then put into a package: an *integrated circuit*, IC or chip. Initially these contained only a few transistors but the number grew rapidly so that each IC could perform more complicated functions.

The growth in the number of transistors on an integrated circuit has roughly doubled every two years since 1965, fulfilling a law proposed by Gordon Moore. *Our World in Data* have a beautiful plot to illustrate this in their article on What is Moore's Law?. Another way of expressing a factor of 2 every 2 years is a factor of 1000 every 20 years. The early microprocessors of the 1970s had a few *thousand* transistors, which had risen to a few *million* in the 1990s and to a few *billion* in the 2010s; the largest processor in the plot from Our World in Data has nearly 50 billion transistors. Astonishing! Not all ICs are so large; many have fewer transistors. Analogue ICs, such as amplifiers, typically have many fewer transistors because it is much harder to shrink analogue functions than digital ones. And, of course, not every computational task requires a massive processor.

The cost of an IC is roughly proportional to its area so smaller transistors reduce the cost. To set the scale, a human hair is roughly 100 µm in diameter or 100 000 nm, which is 10 000 times the width of a modern transistor (10 nm). The wavelength of visible light is around 500 nm, to give another reference.

Modern supercomputers are designed in a different way to Atlas. Rather than have a single, extremely powerful processor, they have a large number of processors to gain their overall performance. As an example of the cost, the current computer for the Met Office will have an overall cost of up to £1.2 billion over 10 years.

Microprocessors enabled the development of microcomputers. The Apple II, Commodore PET 2001 and TRS-80 were launched in 1977 and introduced microcomputers to consumers; earlier models had been aimed at hobbyists and businesses. The IBM PC followed in 1981 and its operating system, PC DOS, is the ancestor of Microsoft Windows. We now call the products personal computers rather than microcomputers.

Developments in integrated circuits were reported on TV at the time, notably in *Tomorrow's World* and *Horizon* on the BBC. The 50 Years of Horizon collection includes Now the chips are down from the 1977–78 season, which 'examines the rise of the microprocessor and asks if automation presents a problem for the future of British industry'. The video looks ancient but many of the questions are still relevant.

If you look inside a personal computer – desktop or laptop – most of the electronics resides on a *motherboard*. This clearly includes the microprocessor and its different forms of memory but other *peripherals* might be less obvious. A *real-time clock* keeps running even when the rest of the computer is turned off, for example. A microcontroller integrates all of the peripherals required along with the CPU and memories.

Microcontrollers were developed to simplify electronic control systems for products such as photocopiers but have since spread into almost everything electronic. A microcontroller *controls* the system in which it is embedded rather than just processing data: the name comes from its function in the system rather than anything special about the integrated circuit itself. It reads in signals and sends outputs to other parts of the overall system. Many microcontrollers have become more powerful, like microprocessors, but smaller devices have also been developed so that they can be included in cheaper products that do not need extensive processing power. Your microwave oven does not need a supercomputer inside it!

RS (a leading distribution of electronic components) have A Complete Guide to Microcontrollers that explains the difference between microcontrollers and microprocessors.

3 Microcontrollers in the world

Let's look at a range of embedded systems that use microcontrollers, starting in the kitchen.

Microelectronics in the kitchen

How many of these products rely on (digital) electronics?

Look at these products in turn.

- **Kettle** – no electronics at all, as far as I know. It switches off when it boils and the mechanism is mechanical. (You can buy fancier kettles with electronics if you wish.)
- **Toaster** – an example of the change from mechanical control to electronic control. We learned in O-level physics that toasters were controlled by bimetallic strips, which bent as they heated up and switched the toaster off when they reached a certain temperature.

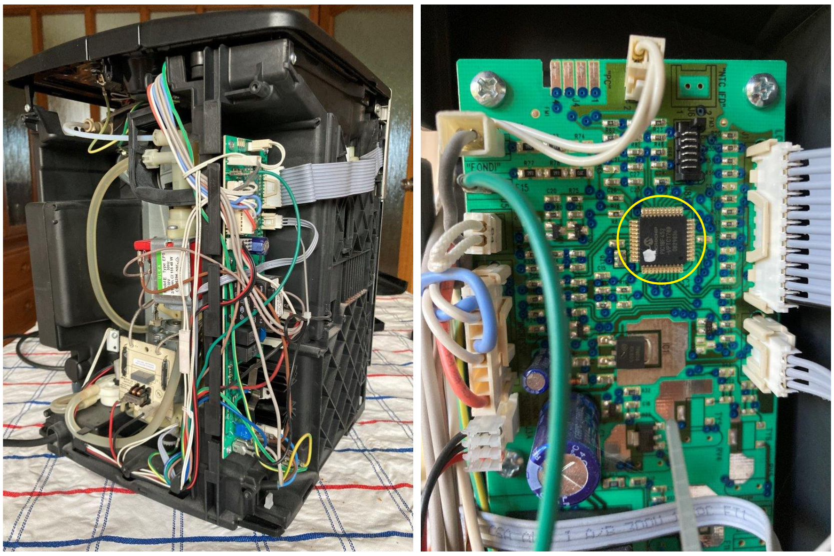
  Now they contain electronics, although this doesn't need much more than a timer. Special timer ICs have been available for many years and an old toaster of ours contained one of them, but it might now be cheaper to use a small microcontroller instead like the 6-pin example that I mentioned earlier.
- **Coffeemaker, breadmaker and microwave oven** – all require something like a microcontroller to manage the sequence of operations and perform them for the correct time. Other examples include washing machines, ovens and so on. Features like delayed start for a washing machine used to be expensive but are trivial with electronic control.

- **Radio and cordless phone** – both contain specialised electronics for their main functions but may also have something like a microcontroller for the user interface: buttons, display and so on.
- **Blender** – an interesting example. This product is ancient and predates microelectronics in the kitchen. The motor is the sort that I learned about in O-level physics: it has only a single speed, it is inefficient, noisy and creates sparks that generate ozone. The motor is also simple in the sense that it needs only to be connected to power to function.

  An up-to-date blender would be different, except perhaps the very cheapest. The motor would be more efficient and quieter with variable speed. It would also require electronics to drive it: modern types of motor do not work if they are connected directly to the power. They need transistors to switch currents in the correct sequence, managed by something like a microcontroller.

Most rooms in a house could provide similar examples.

Inside the coffee maker



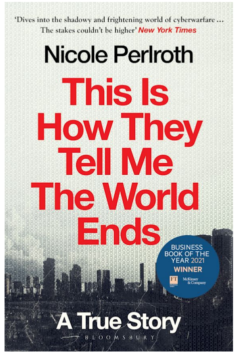
The microcontroller is a Microchip PIC18F452 with 40 pins. It includes an analogue-to-digital converter (ADC) and several timers. These functions are commonly found in microcontrollers, as we shall see later.

The connector at the top with four copper stripes may have been used to program the microcontroller. Scratches show that it has been used although it is not connected in the finished product.

Internet of Things

The name was coined by British entrepreneur Kevin Ashton in 1999. The basic idea is a network of connected, 'smart' devices that replace traditional ones. This is a major application of embedded systems.

Example: domestic heating from tado°.



All manufacturers of microcontrollers have developed products aimed at IoT. Many possible products have been suggested. One of the earliest was the *internet refrigerator*, which would order a bottle of milk when it detected that you were running low. For example, Samsung sell a Smart Family Hub fridge–freezer. It has internal cameras, which you can use from your phone to find out what is inside, and it can alert you when items approach their sell-by dates. This does not come cheap: £3500.

A more common example is the smart meter for electricity or gas. I am not sure how 'smart' they are: I had one installed in November 2020 and the supplier was unable to calculate a correct bill for a year! It is almost impossible to read by hand and I got more useful information from the mechanical meter that was in the house thirty years ago.

Security is a serious issue for the IoT. The stuxnet attack on Iran showed the extent of damage that can be done to networked machinery. Another examples is a hacker taking control of trams in Lodz (2008). 'Smart' home systems may provide a route for hackers into home networks and the devices connected to it. Keep your software updated! If you wish to be scared by the consequences of cyberattacks, you might like to read Nicole Perlroth's book This Is How They Tell Me the World Ends.

You should also be concerned about privacy. The author of an article in The Conversation entitled Internet of Things: tech firms have become our digital landlords – but people are starting to fight back found that "using Alexa's voice command triggers 246 contracts that we have had to accept in order to use it".

An obvious question is whether artificial intelligence (AI) will spread as far as the sort of systems that I describe in this talk. The sort of AI that makes the headlines, such as ChatGPT, requires immense computing power and will not be available in individual user's hands for many years. However, smartphones already contain neural processors, which are another form of AI and are used for tasks such as the identification of faces in photos. According to the Wikipedia page for the A17 Pro, the most recent processor for iPhones at the time of writing (2023 October), 'The 16-core Neural Engine is now capable of 35 trillion operations per second', while the Apple press release claims that it powers 'features like autocorrect and Personal Voice in iOS 17'. If you wish to learn more, u3a now offers subject advice on Artificial Intelligence.

4 What's inside a microcontroller?

Look back at the Atlas system. A microcontroller is clearly constructed on a quite different physical scale but its main components are similar. The key features are:

1. central computer (central processing unit, CPU)
2. various forms of memory, including the tape drives on the upper floor

3. devices for input and output (paper tape, punched cards, printers)
4. operator's consoles for overall control

How do these compare with a microcontroller?

Example of MCU: NXP LPC800, introduced 2017, cost from 30p

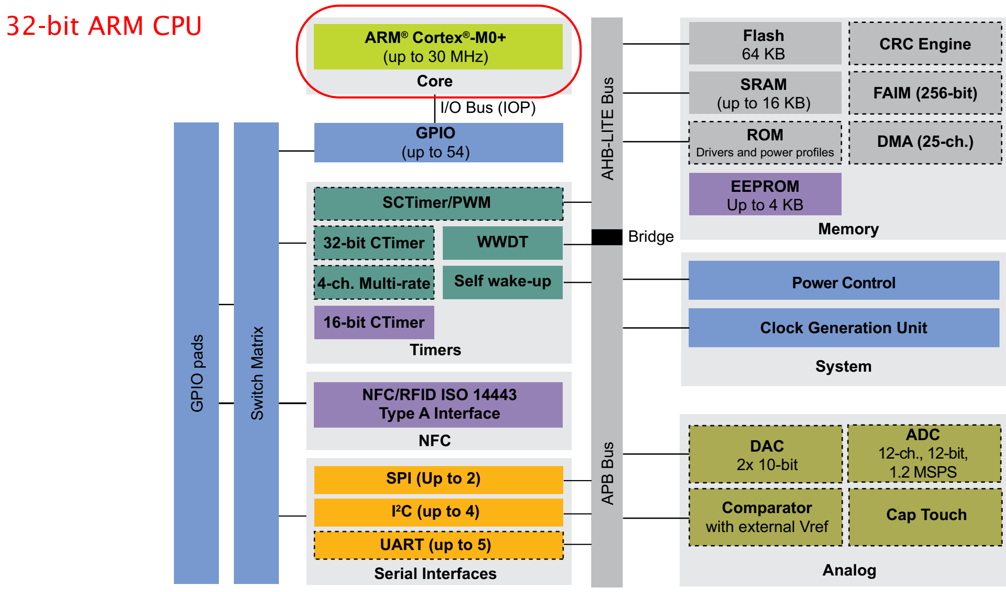


Image from LPC800 Series Low-Cost Microcontrollers (MCUs).

This is a simple (!) product, based on an ARM Cortex-M0+ core. The family includes devices with different peripherals and sizes of memory with prices from about 30p. The first four features match closely those of Atlas.

1. The **CPU** (Core) is an ARM Cortex-M0+, the simplest processor produced by ARM. It operates on 32-bit words.
2. The microcontroller has two main forms of **memory**:
   - **Flash memory** holds its contents permanently (non-volatile), even without power to the chip (until deliberately erased). It holds the program and any supporting data, such as identification numbers. This is similar to the memory used in storage (SD) cards, phones and increasingly in personal computers. The LPC800 series has 4 KB to 64 KB of flash memory.
   - **Static random access memory** (SRAM) holds data only while power is applied to the chip (volatile). It is used for information required while the chip operates – the results of calculations, data received and so on. The LPC800 series has 1 KB to 16 KB of SRAM.

   Two types of memory are needed because it is difficult to write to flash memory: the process is slow and uses a lot of power. Newer types of memory come closer to the ideal of being both fast and non-volatile. Texas Instruments uses ferroelectric random access memory (FRAM) in most of its new MSP430 microcontrollers, for example, even the cheapest models. Ferroelectricity is similar to ferromagnetism but the materials have a permanent *electric* polarisation rather than magnetic polarisation. The direction of the polarisation is used to store a binary value (0 or 1).

Note how little memory is available. The units are KB for kilobytes (binary units of $2^{10} = 1024$ rather than 1000 to be precise). A personal computer has several GB (10^9) of RAM and perhaps TB (10^{12}) of hard drive.

3. Simple input and output takes place through **general purpose input/output** (GPIO) pins. These convey digital values: a voltage near zero (ground or earth) implies a binary value of 0, while a voltage near the power supply implies a 1.

4. The **system** block provides overall control.

   - The chip must start up in a well-defined state. (My set of Christmas tree lights has several functions and starts up in one of these at random when you turn it on. Clearly this would be unacceptable for almost any other product.)
   - Power is turned off for parts of the chip that are not currently being used to save energy.
   - The clock sets the speed of the CPU and keeps different parts of the chip synchronised. The clock in a personal computer typically runs at a few GHz (a few times 10^9 operations per second) but around 10 MHz (10^6) is more typical for a small microcontroller.

These are the familiar parts but the microcontroller has many other functions as well. These units are called **peripherals.** The name makes them sound unimportant but in fact you usually select a microcontroller that has the peripherals you need for an application.

5. **Timers** are a surprisingly important part of almost all embedded applications. Take a washing machine as an example.

   (a) Each phase of the washing cycle – washing, rinsing, spinning – has to continue for a particular time.

   (b) The drum of the machine should rotate at a particular speed, which must therefore be measured – the number of revolutions per minute (or per second).

   The microcontroller uses its timers to set the duration of activities that it controls and to measure the time of inputs, such as the rotation of the drum.

   A timer is a vital part of an operating system so Atlas must have had timers of some sort.

6. **Serial interfaces** allow the microcontroller to communicate with other devices. The interfaces here are typically used to connect other chips but a UART could drive a serial or COM port, which used to be found on every Windows PC.

   Communication was not significant when Atlas was new and programmers would run jobs by loading cards or tapes. Time-sharing became important during its lifetime and programmers would use teletypes or video display units, so electronic communication rapidly became essential.

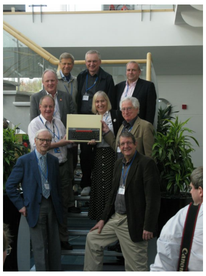
7. The **analog** block converts analog signals – typically voltages – to digital numbers and *vice versa.* For example, a microphone converts sound into a voltage that varies continuously with time. This is amplified and applied to an **analog-to-digital converter** (ADC), which converts the voltage into a number that the microcontroller can process. The output from

almost every sensor, such as the accelerometer in the student project, is treated in the same way.

A **digital-to-analog converter** (DAC) does the opposite, and produces an output voltage proportional to a number. These are much less common because a simpler, digital approach called **pulse-width modulation** can usually be used instead of a true analog output.

5 ARM

ARM – Brief history

From  to 

to 

- Acorn Computers founded 1978; launched BBC Microcomputer in 1981.
- ARM RISC processor family introduced 1985; taken up by Apple and DEC.
- ARM was spun off in November 1990 to focus on the **design of processors**.
- The Japanese SoftBank Group bought ARM for £24 billion in 2016.
- Some shares in ARM are now traded on the US NASDAQ stock market, 2023.

Creators of BBC Micro (Wikipedia, 2012); Nokia 5110 (1998); Apple M1 chip, 2020.

- Left image shows the creators of the BBC Micro: (clockwise from top left) Hermann Hauser, Andy Hopper, Christopher Curry, Sophie Wilson, David Allen, Chris Serle, David Kitson, Chris Turner, and Steve Furber at the BBC Micro 30th anniversary in 2012. Photograph on Wikipedia.
- Middle image shows the first mobile phone that I bought: a Nokia 5110. It is significant because this was the first family of phones to be based on a processor from ARM. Mobile devices rapidly became a huge business for ARM and I believe that their processors are now in almost every such product.
- Right image shows MacBook Air from Small chip. Giant leap. introducing the M1 system on a chip (SoC), Apple Computer UK. ARM have come full circle, back to computers where they started (as Acorn).

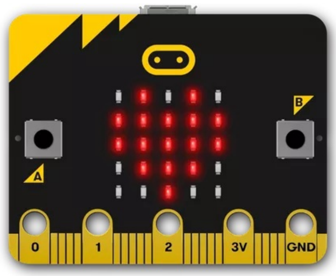
ARM was founded in 1990, over 30 years ago. It grew out of Acorn Computers, best known for the BBC Micro computer. They decided that they should develop their own processor for a more powerful computer, which became the Archimedes, and this led to the Acorn RISC Machine or ARM. A long story, including collaboration with Apple for the Newton hand-held device, led to a separate company called ARM whose primary business is the design of processors. They do not manufacture chips themselves but licence the design to companies who build their own chips around them. Their headquarters is in Cambridge, England.

Softbank attempted to sell ARM to Nvidia, a US company best known for graphics processing units (GPUs). The deal faced opposition from other companies and regulators because it might give Nvidia unfair influence over a large part of the digital industry. The takeover was abandoned on 2022–02–08, after which it was expected that ARM would be floated on a stock exchange. This would probably be the NASDAQ in the USA rather than in London, where it was previously listed. An announcement of an initial public offering was made on 2023–04–29 and the IPO itself took place on 2023–09–14, valuing ARM at US$54 billion. ARM released an article Reflecting on the Arm IPO.

6 BBC micro:bit

From the BBC Micro to the micro:bit

The **BBC Micro** (1980) was a typical *microcomputer*: a standalone machine on which you wrote and ran programs, although other equipment could be plugged into it.



The **BBC micro:bit** (2016) is a single-board *microcontroller*: programs are written on a separate computer and downloaded; it can be built into a larger system.

Micro:bit call this **physical computing** – engineers would say an **embedded system**. Photo of BBC Micro by Stuart Brady on Wikipedia.

Many other types of single-board microcontroller are widely used, including:

- Raspberry Pi, considerably more powerful than the micro:bit, including an operating system
- Arduino, much loved by hobbyists and students, with a vast range of projects on the web (many of doubtful quality)
- ARM mbed, a more professional platform on which the micro:bit is based

History and philosophy of the micro:bit

- Developed in collaboration by the BBC with Microsoft, Lancaster University, Farnell (manufacturer and distributor), ARM and many others.
- **Physical computing** involves interacting with the physical world in addition to a screen. You take in data from light, motion, or temperature sensors, and control devices such as motors, speakers, and lights. Physical computing is fun, engaging, and makes learning to code easier and more understandable. (Quoted by Micro:bit from MIT.

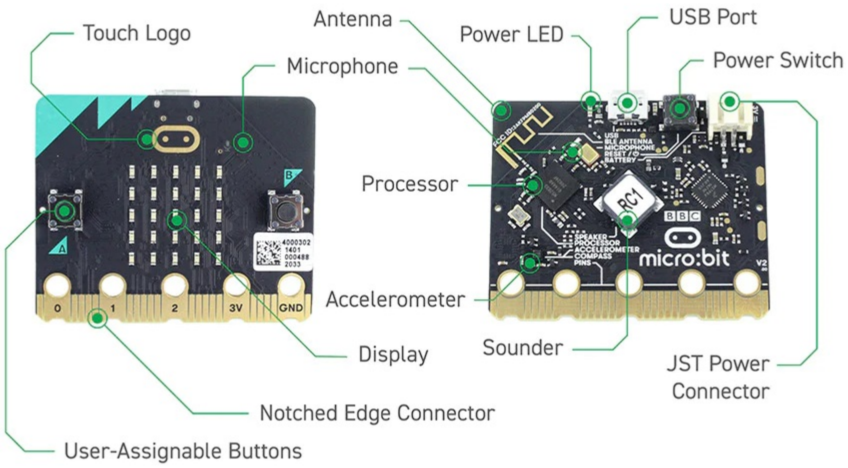
- Every Year 7 student in England and Wales, Year 8 in Northern Ireland and S1 in Scotland (all aged 11–12) received a micro:bit from March 2016.

- Can be programmed using blocks in Microsoft MakeCode, which we shall use.

  Traditional text-based languages are also available, notably MicroPython.

- Led by the Micro:bit Education Foundation since 2016. The micro:bit is used worldwide and the British Council is a partner.

Features of the micro:bit



All this on half a credit card! The two buttons are labelled A and B. The processor includes a temperature sensor. The USB interface chip (right) is not labelled. Image from Kitronic.

Here is a quick tour of the features, starting on the front. The micro:bit web site has several pages with further details, starting with an Overview.

- Buttons – labelled A and B, which we shall use in many programs.

- LED display – 25 LEDs in a 5×5 array, which we shall also use frequently.

- Pins along the bottom – also called GPIO for general purpose input and output, used for connecting other electronics to the micro:bit. We shall probably not use them.

- Touch sensor – the golden micro:bit logo at the top can detect a finger touching it, like the controls of many products nowadays (the touch screen of a phone is a far more sophisticated version of this sensor).

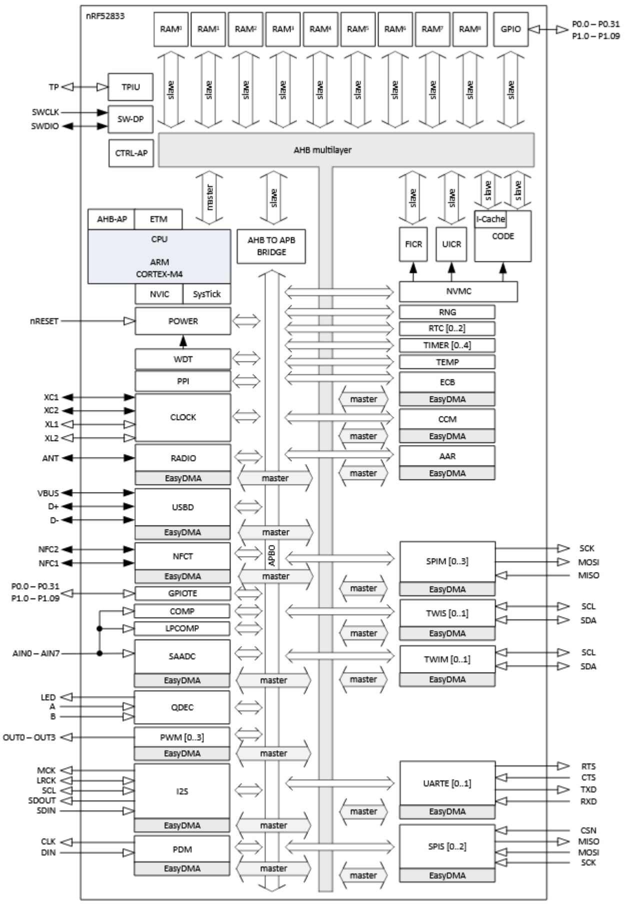
- Microphone – the microphone itself is on the back of the board and receives sound through a small hole in the printed circuit board (PCB). An LED illuminates when the microphone is active.

- USB connector – the micro-USB socket is used to connect the micro:bit to the computer on which you have constructed your program so that the code can be downloaded into the processor on the micro:bit. Two LEDs on the back show when the interface is active.

- Reset button – mainly used to put the micro:bit into pairing mode for Bluetooth. It can also reset (restart) the processor if the program fails and put the processor into a low power mode.

- Radio antenna – required for Bluetooth to program the micro:bit if you do not use USB. It can also be used to communicate (more simply) with another micro:bit.
- Accelerometer – detects whether the micro:bit is being accelerated in any direction, being shaken for example. This includes the acceleration due to gravity, which enables you to find the orientation of the micro:bit. It sounds complicated but is remarkably easy to use in a program. A smartphone uses a similar chip to control the orientation of the display – landscape or portrait – for example.
- Compass – measures the magnetic field and therefore acts much like a traditional compass. It is integrated into the same chip (integrated circuit) as the accelerometer.
- USB interface chip – to the right of the speaker, not labelled; required because the processor itself does not include the complicated hardware required to communicate over USB.
- Temperature sensor – gives the approximate temperature of the processor, into which it is embedded. This is higher than the temperature outside, depending on how hard the processor is working and how much heat it generates.
- Processor – Nordic Semiconductor nRF52833 processor. The computational features are an ARM Cortex-M4F core running at up to 64 MHz, 128 KB RAM and 512 KB flash memory. It also includes a radio for Bluetooth and other protocols used in wireless networks of devices, such as Zigbee.

The processor is far more sophisticated than the LPC800 shown earlier. Here is its block diagram. You can see why I did not put it on a slide!



All this would cost you about £3 for the chip in bulk. Amazing. As you can see, the processor core (the CPU) is a tiny part of the integrated circuit: the 'peripherals' dominate the block diagram and (almost certainly) the silicon chip itself.

The nRF52833 includes an interface for USB. I do not know why a separate chip is used on the micro:bit. Possibly this is for compatibility with the earlier version or maybe there is insufficient memory on the nRF52833 for all the code.

7 About this course

Prospectus

Here is the outline of the practical sessions.

1. **First program: Hello!** – a simple program to get familiar with the MakeCode application (or web page) for constructing the program and with downloading the program to the micro:bit with USB or Bluetooth.
2. **Handling inputs** – responding to the pushbuttons, the simplest type of input.
3. **Variables** – data that change while a program runs, which we shall use to construct a dice with the traditional patterns of dots.
4. **Radio** – micro:bits can send short messages to each other, which we shall use to simulate a swarm of micro:bits that flash in synchronisation.

Further slides cover:

- Tips for using MakeCode, text-based languages as an alternative to blocks, and what goes on 'behind the scenes' of your program
- Functions and how to use them, a vital tool for larger programs

Information on the micro:bit

A *vast* amount of information is available on the micro:bit. This course is based heavily on the following web sites, particularly the home site microbit.org.

- Home site for BBC micro:bit microbit.org, extremely informative, including:
  - Getting started
  - User guide
  - Let's code
  - Mobile apps for iPad and Android tablets
  - Lots of example projects, some of which we shall try
- Microsoft MakeCode, which we shall use to construct our programs
- MakeCode reference and micro:bit Help and Support
- micro:bit developer community for greater technical depth
- Wikipedia page, with a large number of further links
- British Council working across the world in partnership with micro:bit