**The case for an EduMite "ecosystem"**

Here in Wales from 2022 and in the rest of the UK from 2019+ aspects of computer science are a compulsory component of the state maintained education system. From 3-16 in aspects of CS are taught in both primary and secondary settings. At secondary level, this is largely undertaken by "IT/Computing" specialist teachers (but even here, over 80% of these teachers are actually IT in the sense of business systems not computing / hardware aspects of CS). In primary settings this is undertaken by classroom teachers, 90%+ of which have no formal IT/CS training and in many cases, neither the time nor inclination to undertake personal development in this area (as of course for these practitioners everything else they teach is also a priority).

In this context Scratch is the premier language used across nearly all settings until learners opt for computer science (a minority GCSE with less than 10% of learners opting for). At this point Python and Visual Basic are the languages of choice (even though none of the qualifications actually specify which languages to use).

In primary settings, Scratch is used to develop "Games" and the overwhelming evidence is that formal development of computer science techniques such as decomposition of large problems, abstraction of complexity, iteration, conditional logic and appreciation of data structures is almost none existent. Learners copy block code from the example and tweak things – understanding is limited (by teachers and learners). Most curricula also require aspects of "physical computing" to be developed – so schools use Microbits and LegoMindstorms – however, these are still programmed my "Scratch like" block coding (even though Python / JavaScript is available on these platforms). And, the "physical" devices are often either "black boxed" (LegoMindStorms) or wrapped up in excessive "kid proofing" that all the users have no idea what components they are actually handelling. An additional challenge arises as children (and most teachers) do not perceive the Microbits / LegoMindStorms as a microcontroller / computer – but rather an extension of the PC that they are plugged into – and a "low spec" extension by that thought process. They interact with the tech as though they are "programming" it – similar to a PIC chip. They lose all concept that there is a "processor" in there doing something with their code. To top this off, schools struggle with "cables". In many cases VGA / HDMI / USB cables are attached in a manner that they cannot be removed without "IT" permission, if at all. Plus, cables "go walk about", rendering equipment unusable.

Whilst Microbits where initially given free to schools, they are now £30+ each. Schools then go on to purchase "robots" and other high value items in an attempt to entice learners into engaging with the technology.

Without being overly negative at this point, what has happened is that we have "gamified" the tech, to the point where most people (teachers and learners) see no use outside robots and entertainment. As a result, the pay off for the learners is always less than they anticipate – they can't code "Call of Duty" nor make a VR system – so interest usually wains very quickly. The bar is too high, and no one lives up to it.

Given that (primary) teachers struggle with basic electronics (circuits, components, diagrams etc) and have no formal CS training, what has exploded onto the market is range of "kit" that is specifically designed for educational markets. The selling point of this kit is that it will come with "lesson plans" and resources to support the teaching. However, what that means is that teachers "learn" how to use the specialist kit but do not develop any transferable skills. Replace the Microbit with an Arduino, Raspberry Pi or a 'mite any they do not know how to proceed. Equally, their skills in Scratch do not translate into any ability to pick up Python and figure it out. A secondary consequence is that learners can not access the kit at home – its both massively too expensive and the resources are all geared up to "teaching".

In practicality, the largest "use case" of these devices is in a primary education setting – both in terms of number of schools ~ 9 times more primary schools than secondary (UK) – but also in the curriculum opportunities to use them.

**Opportunity:**

What is needed is a "tinkering" platform at a price point that is suitable for both users in class and for purchase to use at home, with the ability to utilise "real" components (either on breadboard or through mounting individual components onto small PCBs connected together via crocodile clips) – and with "built in" sensors etc.

**Uses:**

(1) Teach basics of "electronics".
    a. Circuits (series / parallel / switch etc) etc
    b. Components (LEDs, diodes, resistors, piezo buzzers, etc)
(2) Use the above to "**design**", "**build**" and "**code**" solutions.
    a. Traffic lights / pedestrian crossings / control of LEDs
    b. Temperature / humidity / light sensors to turn on / off something
    c. Build a "data logger"
    d. Build "complex" systems – possibly with GUIs?
(3) To teach the idea of "programming" a microcontroller

**Requirements** -- am thinking something in the same form factor as the CMM2. Whilst integrating sensors will increase the BOM costs, the device needs to be "useful" out of the box, from power on – otherwise these components need to be purchased by schools anyway.

(1) **Self-contained (as possible)** (Not physically tethered to PC for programming) (Serial Bluetooth module) (But direct connection possible for alternative connection / flashing firmware)
(2) **Integrated screen** for feedback / display (IPS panel?) (Small is OK, on the front)
(3) **Integrated "sensors"** (hardwired to PicoMite i/o)
    a. LDR (exposed on front / top)
    b. Temperature / humidity DHT22 or equivalent (vented case ??)
    c. Sound sensor (detect clips / loudness) (vented case ??)
(4) **Integrated LEDs** (hardwired to PicoMite i/o)
    a. X3 hardwired LEDs (Red, Amber, Green)
    b. WS2812B (strip ??)
(5) **Integrated audio (**hardwired to PicoMite i/o) on board – buzzer / speaker?
(6) **SD card**
(7) **Power** (connect to the back)
    a. Run from USB wall power
    b. Run from external battery pack (only needs to run for an hour or so for teaching, longer of course for sensing / logging)
**(8) Case/enclosure (CMM2 sized?)**
    a. Power "on" LED (Front) / toggle power switch
    b. With the display panel visible (Front)
    c. With the LEDS (WS2812B) visible (Front)
    d. With "buttons" (hardwired / digital i/o) (TOP?)
        i. 1 momentary switch
        ii. 1 spst toggle switch
        iii. 1 rotary potentiometer (control WS2812B for example)
    e. With banana plug sockets (hardwired to several pins, voltage and ground) (TOP) (to allow connection to leads with croc clips on)
    f. With I/O breakout via a standard header (on the "back") for the un allocated pins / power / ground.
    g. Access to the "flash" / "reset" functionality from back (momentary switch / USB?) (so teachers don't need to take the box apart to flash it).
**(9) Protection**
    a. Polarity protection on exposed i/o pins (?)

Could the PicoMite be fitted into a socket so that it could be swapped out in the event of something catastrophic (ie a child) happening to it?

**Plus, a range of educational resources / lesson plans etc (on me / teacher colleagues)**

**Nice to have (wish list if possible):**

(1) Real time clock (maybe an "optional" plug in module?)
(2) Integrated LiPo charger / battery (again, an option?)
(3) Some way of getting physical output / thermal printer (I can dream)

**Not needed:**

(1) PicoMite W functionality – schools will struggle with SSID / Password / IP access. Remote access provided by Bluetooth.
(2) PicoMite VGA functionality – as given in the blurb – cable swapping not going to happen – the point here is to be as "stand alone" as possible.

**Separately (but part of the ecosystem)**

(a) Break out box compatible with the standard header (not breadboard)
    a. More banana plugs
    b. Switches etc
    c. With a "patch panel" to wire them to 'mite pins – like Engima plug board!!
(b) I/O header to standard breadboard adapter
(c) Standard components on small PCBs (to make handling easier) - but not over sanitised like https://www.techsoft.co.uk/products/electronics/locktronics - so that they can be connected with crocodile clips to make simple circuits
(d) Modules that connect to either the i/o header or banana plugs:
    a. Things like "traffic light" modules
    b. WS2812B strips / modules etc
    c. Other I2C devices that are compatible with it
    d. Additional sensors / servos / motors etc etc
(e) Physical output – Bluetooth printer (not sure how this would work unless there were two Bluetooth modules?)