

AUTUMN 2015

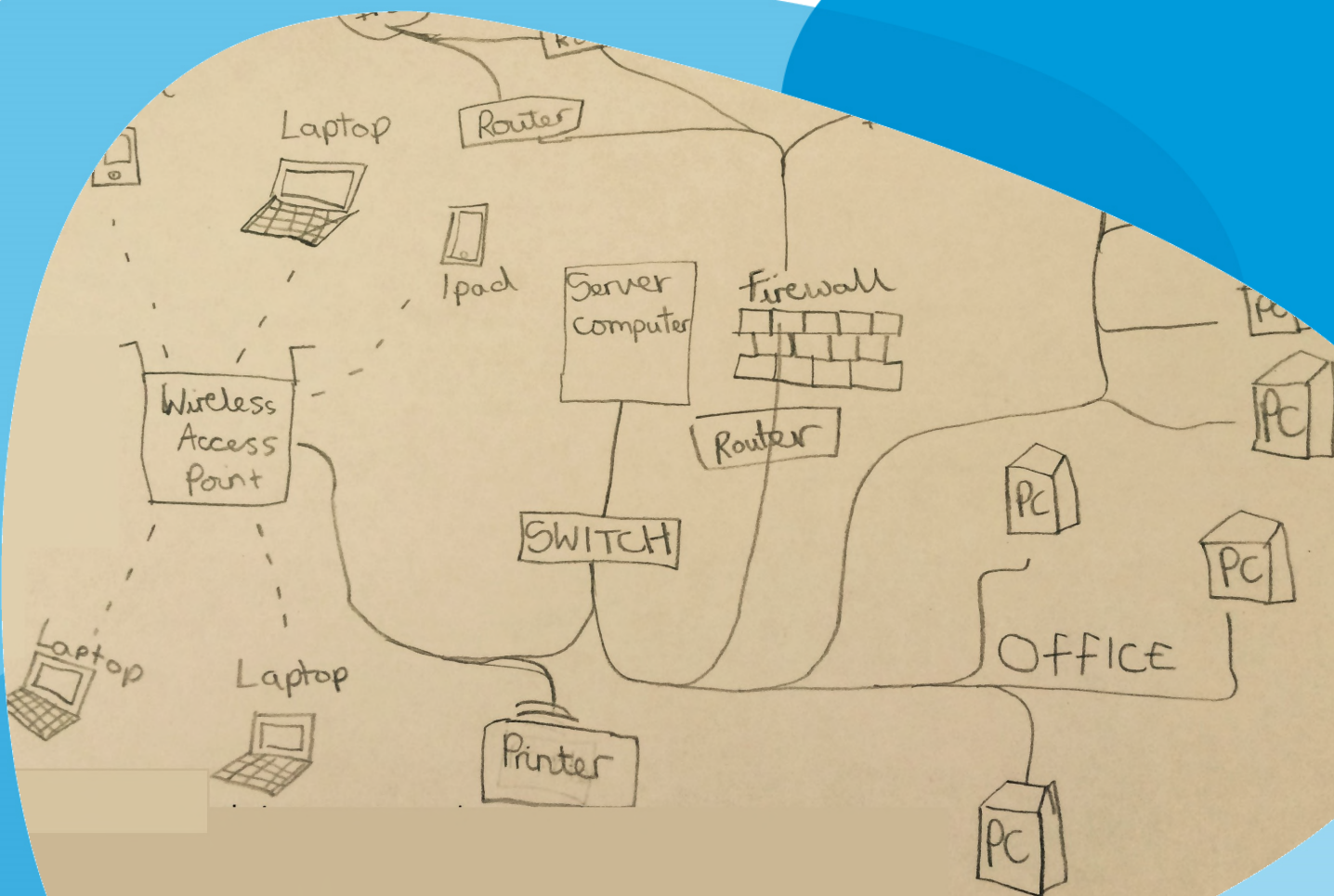
# Computing/ICT User Friendly

## In this issue:

Introducing Blue-Bot and MaKey MaKey

Robots with sensors and computational thinking

Dispelling the myth of the digital native



# Contents

Editorial	1
Introducing Blue-Bot	2
Introducing the MaKey MaKey	4
Robots with sensors and computational thinking	6
Attitudes to computing: dispelling the debilitating myth of the digital native	10
Upcoming courses	11

# Editorial

Welcome to another edition of *Computing/ICT user friendly*.

Firstly let me introduce myself as the new curriculum update editor. I joined Hampshire's Inspection and Advisory Service (HIAS) just over a year ago, after many years as an ICT advanced skills teacher. I currently combine teaching computing at three schools, working for Hampshire and writing computing resources, which I publish on <http://code-it.co.uk>. I was involved with drafting the National Curriculum for computing through the British Computing Society (BCS) and Computing at School (CAS). My wife is a teacher and I have two children and a very elderly cat.

I have changed the name of the curriculum update to *Computing/ICT user friendly* to reflect the name change of the subject, but kept ICT in the title as schools need to continue to include a strand of ICT, although it is sometimes called digital literacy. The National Curriculum talks about ensuring that all pupils are responsible, competent, confident and creative users of information and communication technology. This edition is mostly about computing science, but future editions will include articles about digital literacy, e-safety, and information technology as well.

The image on the cover is by Grace in Year 6 at Otterbourne Primary School and reflects work on how a school network works. There will be more information about this in the next edition.

I am always looking for new articles on a wide range of computing topics so please contact me if you would like to share your journey with Hampshire.

Both Sue Savory (County Inspector/Adviser for ICT) and I can be contacted to help you make the most of computing in your school. (See the inside back cover for contact details.)

## Phil Bagge

Inspector/Adviser for Computing

## Introducing Blue-Bot

Most computing co-ordinators will have heard of the programmable toy Bee-Bot®. However, in January 2015 TTS released the Blue-Bot®. So what is Blue-Bot? What can it do? Do we need one in our school?

At first glance the Blue-Bot looks like a transparent Bee-Bot. It is very similar in shape and has all the same buttons that you will be used to on the Bee-Bot. The Blue-Bot can be controlled in exactly the same way as Bee-Bot. The arrows on the device itself can be used to program sequences of instructions, moving the Blue-Bot, forwards and backwards, turning left and right and pausing at a given time in the sequence. The Blue-Bot is the same size as the Bee-Bot and moves in steps of 15cm like the Bee-Bot, so you can use all the old Bee-Bot mats you have made, or bought, in the past. It also charges in the same way as the rechargeable Bee-Bots that have been around for the last few years. This means if you have a Bee-Bot charging dock you could also use it for any Blue-Bots you may purchase in the future.

While the Bee-Bot is made of yellow and black plastic, the Blue-Bot is transparent so you can see the circuit boards and other components inside, which will help children to start to



develop some understanding of how it works. However, the real difference with the Blue-Bot is that it can now be programmed using an app on an Android, or iOS device, such as an iPad. TTS has promised a PC and Mac version eventually, but these are not yet available. One thing to be aware of though, is that the Blue-Bot is controlled via Bluetooth and will only work via Bluetooth version 3.0/4.0+EDR. iPad3 or later and iPad minis are all fine, but you will not be able to control it using an iPad2.

When pairing a large number of iOS or Android devices to a set of Blue-Bots it could easily get very confusing, with children unsure which Blue-Bot their device is controlling. Therefore, when pairing a class set of Blue-Bots with a set of devices, do it in the following way. First

switch each of your devices on (without launching the app). Next turn all of your Blue-Bots on. Then open the app on the first device. It will search for a Blue-Bot. When it locates one it should automatically pair with the Blue-Bot. The eyes of the Blue-Bot it has paired with will turn blue. Pass that Blue-Bot and the device to a group of children. Then you are ready to pair up the next device with a Blue-Bot. If the Blue-Bots are not pairing to your devices, ensure that Bluetooth is enabled in the device settings.

The app, like the programmable toy, is great because much of it is flexible and open-ended. Unlike some of the game type programming apps, such as the Bee-Bot app or Lightbot, you can set your own problems, allowing the teacher to ensure progression and appropriate

levels of challenge for all the children in a Key Stage 1 class. Any of the Bee-Bot mats that TTS sell are already represented in the app. You can also photograph your own home-made maps and upload these into the app.

The app itself comes with limited functionality when you first install it, but additional functions are opened up once you first pair your device with a Blue-Bot. When the full functionality has been enabled you can control the Blue-Bot in the open-ended explore mode. You are able to control step-by-step for your very youngest learners, but most Key Stage 1 children will be starting to write short algorithms from which they can create short programs in the basic program mode. Children can also explore repetition (loops). There is also a challenge mode where children will solve pre-set problems. While this is obviously less open-ended you can tailor it to the needs of your learners to some extent as there are multiple levels in each challenge. The challenges include getting from A to B, getting to a destination avoiding obstacles, getting to a destination with some of the instructions having been removed and interpreting random instructions.

The Blue-Bot can also be programmed using 45 degree turns, although if using any kind of mat the distance between the centre of one square and another diagonally adjacent will

not be 15cm so this feature may not work as well as you had hoped. Also, at Key Stage 1 children's understanding of angles is limited, so I usually find the right-angle turns that the Bee-Bot was capable of are challenging enough. To me the challenge will always come from the complexity of the problem you set the children. If they are getting it right first time, it is too easy and they are not learning as much as they should be.

So do we need Blue-Bots at our school? They are a little more expensive than Bee-Bot (currently Bee-Bot retails at £49.95 + VAT and Blue-Bot retails at £69.95 + VAT on the TTS website). If you already have lots of Bee-Bots or other programmable toys then I would not rush out to buy them. The computer science elements of the Key Stage 1 computing curriculum can be taught very well using Bee-Bot and there is no need to duplicate. However, if you need to buy more programmable toys or replace damaged programmable toys, and you have iOS or Android devices, they may well be worth spending the extra money on. They certainly give you a lot more than a basic programmable toy.

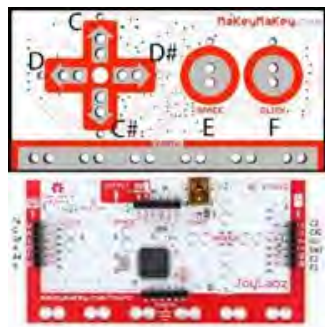
### Emma Goto

Senior Lecturer Teacher Development  
University of Winchester

Emma Goto is a former Hampshire advanced skills teacher for ICT. She has had many years' experience of teaching in Key Stage 1 and the Early Years Foundation Stage and has a real interest in developing young children's understanding in computer science. She is currently lecturing in Initial Teacher Education at the University of Winchester.

# Introducing the MaKey MaKey

As educators we all strive for the same goal within each of our lessons, those moments that wow the children and those moments when you see a spark ignited in the imagination of the children. The amazing little device called the MaKey MaKey is one such device that will give you these moments over and over again!



Created by the people at JoyLabz, this remarkable device "is an invention kit for the 21st Century. Turn everyday objects into touchpads and combine them with the internet. It's a simple invention kit for beginners and experts doing art, engineering, and everything in-between" ([www.makeymakey.com](http://www.makeymakey.com)).

I first began using these devices after seeing them showcased at the CAS Wessex Conference at the University of Southampton back in 2013 and over the years have used them for several different reasons and desired outcomes.

Firstly, I have used them as an awe and wonder device to spark the thinking of the

children, to make them think "how is that possible?!" Noticeably so when I first introduced my Year 4s to my new mouse! I explained that my mouse had broken and I had to replace it with a pencil drawing I had made. Amid the laughter and chuckles of the children thinking I was finally mad, gasps were being heard as they began to see me operate my PC by simply touching the mouse's feet and ears! The look on the faces was priceless! But what ensued was question after question of "How is that possible?" This led into a discussion about how it was possible and this tied in nicely with the current science topic of electricity and circuits. This furthered into a discussion about how we could then use the MaKey MaKey in science to test circuitry, instead of the time honoured tradition of lighting a bulb!

Another awe inspiring moment of note was with Year 3. To introduce them to this cool little device, I wired up five children to create a human piano! The children could not believe that they had become the input for the computer! This was met with, again, the "But how?" questions. Amazingly, what this led to was the forward thinking of the children who then began to ask, "Would it work if two people were holding hands?" Well, by the time we had answered the question we had

a daisy chain of 30 children as one massively long input button! From this, we discussed how we could use the MaKey MaKey in our programming ...



... and so my second point for loving these devices. Once the children are hooked and inspired to use these devices there is no limit to what they think and what they can achieve. The Year 3 children decided they wanted to make their own pianos and so set out to do so. The unit of work centred around Phil Bagges' *Music as code* ([www.code-it.co.uk](http://www.code-it.co.uk)). We began looking at musical notes as an algorithm and discussed how

we could turn this into a viable program that allows all ages of children to play a tune. With Year R in mind, we decided that colour coding was best. The pursuing weeks saw the children turning code into colour, a separate colour for each note, and resolving bugs that sprung up in their programming. The children turned their up, down, left, right, space and W key inputs into colour buttons; when I say buttons, I refer to marshmallows! Yes, the children made their buttons out of marshmallows!



Another project to note for excellent cross-curricular links was a talking poster! Traditionally, as teachers we ask children to make posters of topics to show their learning. I made posters with a twist! The concept was to turn our topic learning about the Mayans into talking, interactive posters. Literacy time was dedicated to the written scripts about aspects of Mayan life learnt during topic time. We used art sessions to draw the posters and buttons and, once the posters were under construction, touched on design technology (DT) and science to make springy buttons that

worked an electrical circuit. Once the posters were completed, programming could begin to insert the sound files into Scratch and then be played back via a computer when the buttons on the poster were pushed.



In summary, a very simple input device can spark the imagination and creative thinking of the children; it will spark the creative thinking of the teacher too! It can be used for a very specific goal or can be used more broadly to touch on many cross-curricular links. Best of all ... they are fun!

**Dan Trickett**

Cherbourg Primary School

## Personal biography

My name is Dan Trickett and I have been teaching primary school children for 10 years. I currently work at Cherbourg Primary School where I manage the computing curriculum and oversee the management of the school's network. I have a passion for technology and a passion for how it sparks and engages children and facilitates learning. I love new gadgets and always find a way to incorporate them into my teaching and into my curriculum. I recently completed my training as a CAS Master Teacher for primary and enjoy sharing my passion and expertise in the field of computing and computer science with all.

# Robots with sensors and computational thinking

Robots today are roving Mars, collecting data in dangerous environments, hoovering our floors, lifting patients in hospital, building cars and entertaining us in films, along with uses in surgery, in military surveillance and drones. The Japanese Robotics Association has made projections that the sales of robots will increase by over five times in the next 20 years. They predict that there will be an increasing demand for service and medical robots that could affect the job market for white-collar workers in the future. Hence our children need to be aware of and prepared for this ever-developing technology.

Children love robots, especially when it is something that they are familiar with, such as Lego. The EV3 robots are very responsive and it is extremely rewarding to see the delight on a child's face when she or he runs their first program and the robot actually moves following their instructions. This really inspires them to write more complicated programs to further test their robot.



Playing with a Lego EV3 robot with an ultrasonic sensor. Put your hand out and the motors stop. Put a repeat on and it keeps doing it.

This article goes through the computational thinking to program a robot to use a sensor, in particular an ultrasonic sensor. I like to ensure the children understand what is happening by decomposing the problem, producing an algorithm with a detailed flowchart, then using it to write and design a program for Lego Mindstorms.

## Introduction

At the start of the lesson, I summarise the main aim of the session is to program a Lego EV3 robot with Mindstorms software to use an ultrasonic

sensor. **Step 1** is to programme our robot to move forward until the ultrasonic sensor detects an object and then the motors stop. **Step 2** is to move forward to detect an object, stop, go backwards and then go forwards at an angle. Then **Step 3**, repeat the whole process at Step 2 by the robot continually avoiding multiple objects.

(For the purposes of this article and the constraints of space, I will *only* be discussing Step 1.)

## What is an ultrasonic sensor?

An ultrasonic sensor can measure the distance to an object in front of it. It does this by sending out sound waves and measuring how long it takes the sound to reflect back to the sensor. The sound frequency is too high for you to hear (*ultrasonic*). Children usually volunteer the information that it is used in nature by bats and dolphins to navigate and find their prey. Also ultrasonic (sonar) is used

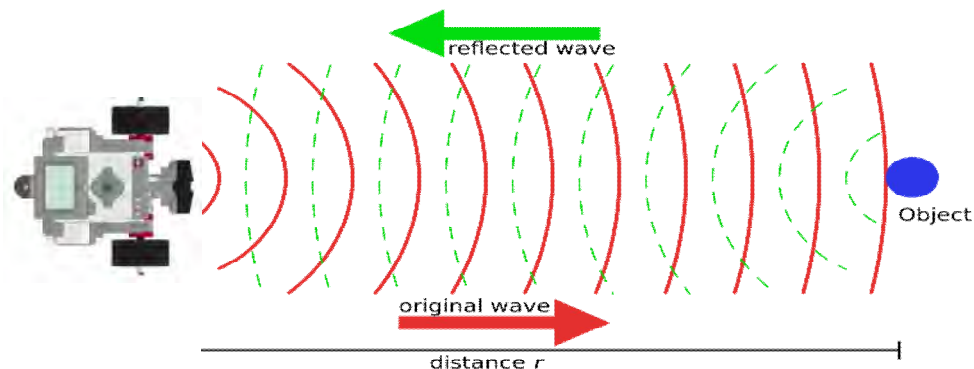


Diagram to show how the EV3 Robot detects an object by using the ultrasonic sensor detecting sound waves.

by submarines and ships to detect objects under the water.

I talk through with the children that the robots start off without any senses. Imagine if we were like this and encourage them to put their *own* hands over their eyes, nose, mouth and ears (see Robot Liz-ii in the photograph below) ... just like our robot. Now we have to programme our robot to have senses too.



Robot Liz-ii (Liz Stoneham) with eyes and mouth shut, ears and nose covered.

## What is an output and what is an input?

The ultrasonic sensor is an *input* device, as the computer receives data or *data goes in* for the computer to give a response. The motors are the *output*, as they move as a result of the ultrasonic sensor sending it information and *data goes out* to turn the motors.

The Lego EV3 robot is prebuilt and the children add the ultrasonic sensor onto the front of the robot. *Outputs* plug into the letter ports – the motors – and *inputs* plug into the number ports and the ultrasonic sensors are *inputs*.

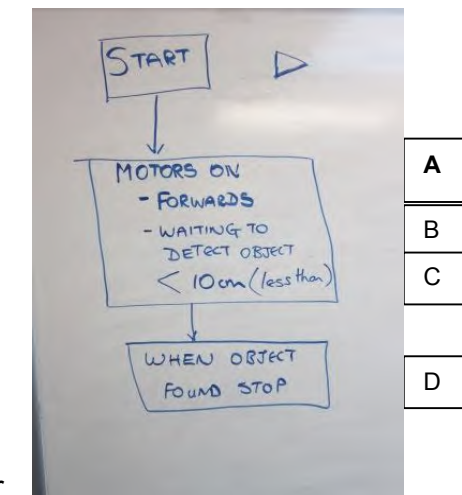
## Step 1: Program robot to move forward, detect object and stop

### Walk through the process

The first aim is to move forward, detect an object with an ultrasonic sensor and then stop. I normally pretend to be a robot and walk through the process step by step. I ask the children to suggest how they think it works and I follow their instructions to the letter, meaning I *only* move when they tell me; I may *not* detect an object; and I might walk through an object or knock it over unless I am told otherwise.

### Flowchart/pseudocode

The robot starts, motors are ON going forward (A) and are WAITING (B) to detect an object less than 10cm away (C). This distance can be varied. When it detects the object at the specified distance the object stops (D). From these instructions, I draw a detailed flowchart on a flipchart,



Detailed flowchart to show detection of object by ultrasonic sensor.

as seen below centre. I like to think that my diagrams verge on pseudocode, providing detail that matches up with the Mindstorms programming blocks to help design our program.

Next we use Lego Mindstorms blocks to match up with the flowchart/pseudocode at A, B, C and D. See diagram on next page.

Download the program and test. Debug any problems:

- is the wiring consistent with the ports on the programming block?
- check programming blocks – is less than (<) and/or the correct distance selected?
- is the ultrasonic sensor put in the correct position – too low or too high?
- are the wires in firmly?

This is the tricky stage of the whole task and may require several testing and debugging stages until the program works. However, once completed, the remaining Step 2 and Step 3 are fairly straight forward.

Using robots is a fun task for the children and many aspects of the computer curriculum are covered without them having any idea that they are programming, including the activities found in the table on next page.

Finally, by introducing the workings and programming behind the behaviour of robots,

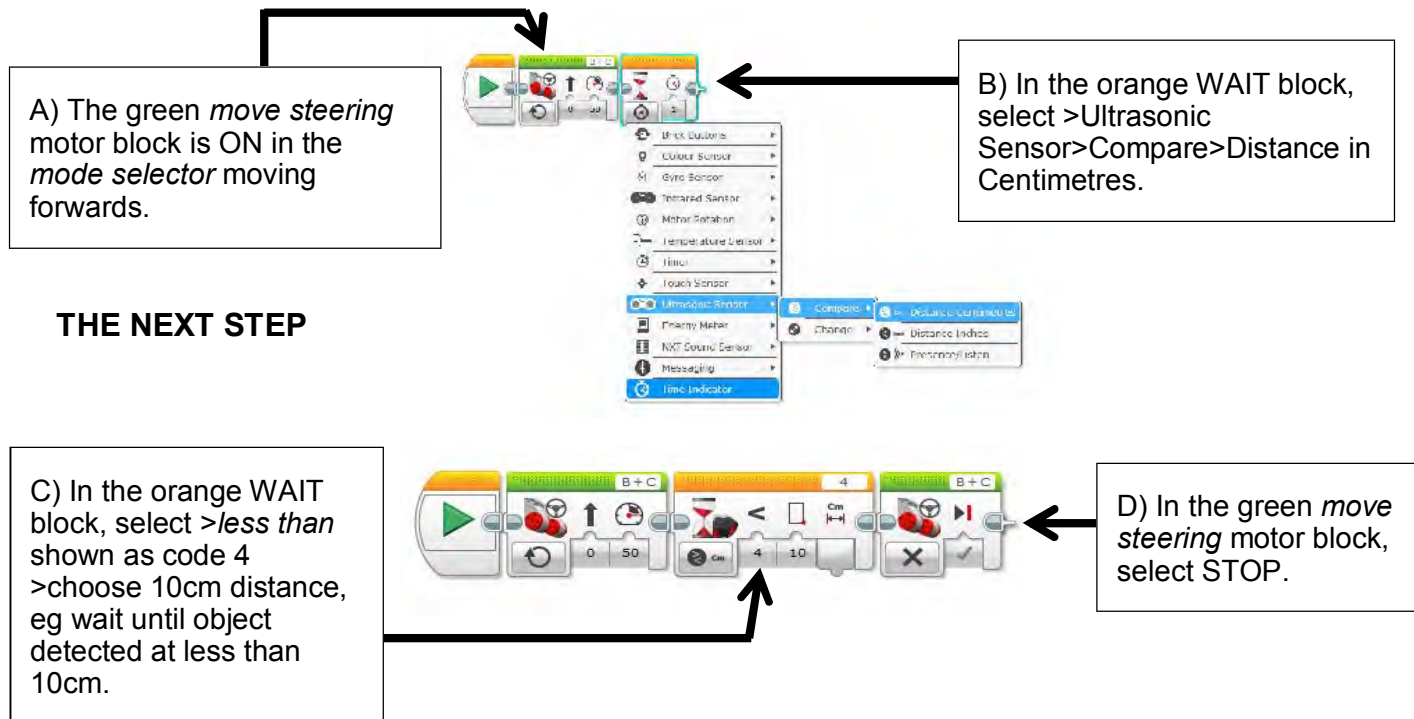


Diagram to show the Mindstorms programming blocks to move motors forward until an object is detected with an ultrasonic sensor and then stops.

will do it and why shouldn't it be our children?

**Robot Liz-ii AKA Liz Stoneham**

Crofton Hammond Junior School

Liz runs robots workshops, lessons and clubs in schools and can be contacted to work with your school at: E-mail: [lizstoneham@yahoo.co.uk](mailto:lizstoneham@yahoo.co.uk).

See Liz's website at [www.robotkidsandspace.co.uk](http://www.robotkidsandspace.co.uk) for more information.

Liz is an IT professional working at Crofton Hammond Junior School and has been running robot clubs for over 12 years.

Activity	Aspects of the computing National Curriculum
Walk through the activity using the ultrasonic sensor.	Solve problems by decomposing them into smaller parts. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.
Write a computer program, test and correct errors to get robot to move correctly.	Design, write and debug goals.
Robot moving.	Controlling a physical system.
Put a series of robot moves together and then repeat.	Use sequence, selection and repetition in programs.
Use motors and an ultrasonic sensor.	Work with various forms of input and output.
Change the speed and direction of the motors and the distance that the ultrasonic sensor detects an object.	Work with variables.

we can take advantage of our youngster's enthusiasm for robots. Hopefully this will

produce a life-long interest and skill that can encourage them to code and program in many

languages in different areas and, possibly, build the robots of the future. After all, someone

## Attitudes to computing: dispelling the debilitating myth of the digital native

In every group of primary teachers I work with I hear that, “Pupils know more about computers than us”. Often this idea stems from the theory of the *digital native*, that people who were brought up with technology are implicitly better at understanding and using it than those who did not.

When teachers say this, it is often accompanied by the statement that “I am just learning alongside the children”. At first glance this sounds very noble. It recognizes that we are all learning and that information exchange can go both ways. However, there are intrinsic problems with this belief and the typical response to it.

The problem with this is that it is at best a half-truth. Pupil access to technology can be very varied in homes from almost identical social backgrounds. Without formative assessment of key skills you will not know what digital literacy skills they are capable of or what they have been exposed to at home.

There is a good research paper on the myth of the digital native here: <https://goo.gl/tBgfKB>.

The wider truth is that there are large swathes of computing that they will not know anything about. Very few have any idea how networks, the internet and the web function or how computing devices are programmed. They do not understand the fundamental computational thinking skills of algorithm and algorithm evaluation, decomposition, abstraction and generalization. They do not appreciate sequence, repetition, selection or variable use, or the resilience and logical thinking that come from computational doing. Even if they are keen users of web resources, few will have inculcated the essential knowledge to develop into safe

digital citizens or to question the veracity of information.

The problem for teachers is that if you do not believe that you have anything valuable to teach because pupils know it all already, this affects what you teach and the time you spend resourcing and planning it.

Teachers and school leaders who believe the myth of the digital native often find excuses to leave computing out of their timetables. When they do teach computing they tend to rely very heavily on unplanned exploration and they justify this by recourse to the shared journey narrative.

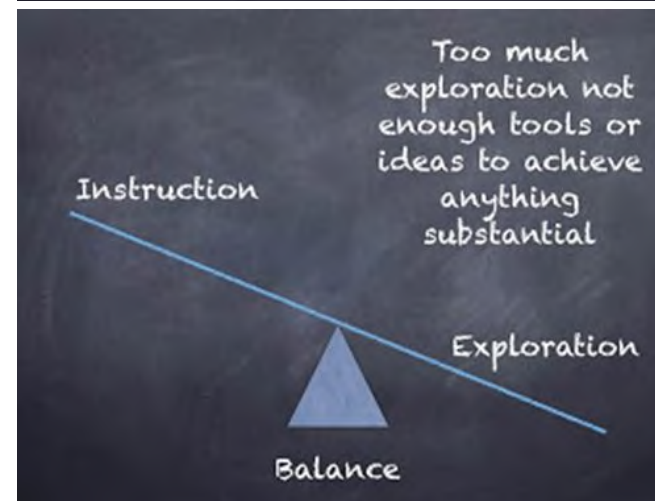
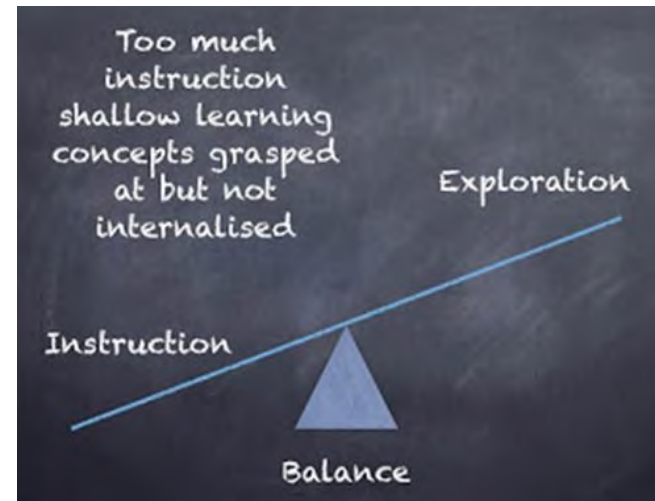
I wonder if they would be happy to learn maths or literacy alongside their pupils in this same woolly way?

Please do not think I am attacking exploration, it is a fundamental part of a good computing lesson but it needs to be balanced alongside instruction in some form<sup>1</sup>. Instruction assumes that there is knowledge and skills worth teaching to pupils.

By way of balance too much instruction without any exploration leads to shallow learning, concepts grasped at but not internalised fully.

We can sum up this fundamental balance like this (see diagrams on opposite page).

In conclusion, as leaders in computing in our schools it is important to challenge the debilitating myth of the digital native and promote curricular that includes instruction and exploration.



For a humorous secondary school look at the myth of the digital native the @codeboom article is well worth a read: <https://codeboom.wordpress.com/2012/07/16/one-does-not-simply/>.

**Phil Bagge**

Inspector/Adviser for ICT

## Upcoming courses

Information on forthcoming computing/ICT courses is available on the Hampshire Teaching and Leadership College (HTLC) website at:

<https://learningzone.hants.gov.uk/learningzone/search.aspx?q=HIAS+ICT>.

<sup>1</sup> The form of instruction should always be down to the needs of the students and the professional judgement of the teacher. Instruction can be as diverse as creating knowledge videos favoured by the flipped classroom model to whole class teaching.

# Contact details

Sue Savory, County Inspector/Adviser for ICT

Tel: 023 9244 1442

E-mail: [sue.savory@hants.gov.uk](mailto:sue.savory@hants.gov.uk)

Phil Bagge, Inspector/Adviser for Computing

Tel: 023 9244 1521

E-mail: [phillip.bagge@hants.gov.uk](mailto:phillip.bagge@hants.gov.uk)

Website: [www.hants.gov.uk/education/hias/computing](http://www.hants.gov.uk/education/hias/computing)

Moodle: [hias.hants.gov.uk/ict](http://hias.hants.gov.uk/ict)