

Digital Devices in the Digital Technologies curriculum (F-10)

Steve Allen
VCAA Digital Coding Specialist Teacher

- A digital system that students can program:
 - create an algorithm
 - accept user input
 - store or process data
 - produce output
- Students work towards Achievement Standards as they demonstrate understanding and application of coding concepts in combination.
- Each subsequent level adds complexity – continuum of learning.

Algorithmic development

F - 2

Linear - one path to take

3 - 4

User input - step through algorithm in response to user

Branching - user input or another condition selects a different set of instructions

5 - 6

Iteration - repeating part of the algorithm a set number of times or until a condition is fulfilled

7 - 8

Functions - discrete group of instructions that are called to action in defined conditions

9 - 10

Modular - reaching out to another set of instructions that have a more specific purpose or focus (methods)

A list, but not a shopping list

VCAA is not advocating any particular device that we look at today – your school context will be the biggest influence

Digital Technologies curriculum is not about devices – approximately 50% can be taught unplugged

If deciding to purchase devices, think about

- your school context
- longevity of the device use
- use in other curriculum areas

Is any other equipment essential to operate your new digital devices? (tablets, computers) How is it powered? (rechargeable/batteries)

Bee-Bot

- The Bee-Bot is a robot with 4 directional keys (forward, backward, turn right, turn left) that can be used to program up to 40 commands
- Useful to develop basic algorithmic understanding in F-2 students
- Limitations – cannot perform branching or looping operations



ProBot

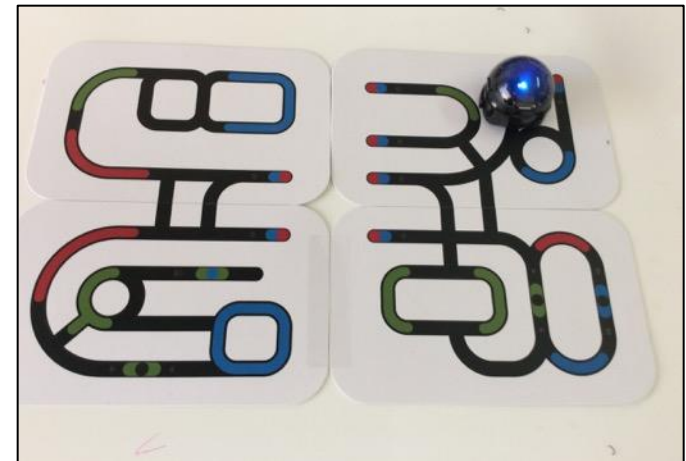
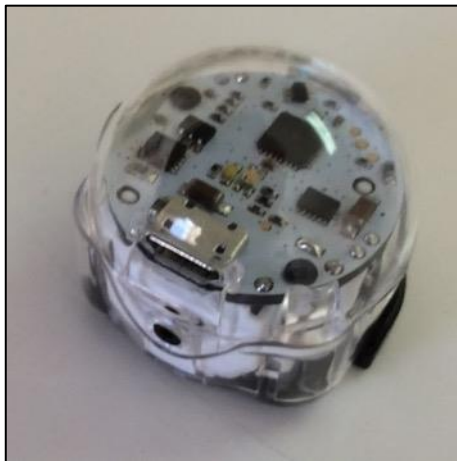
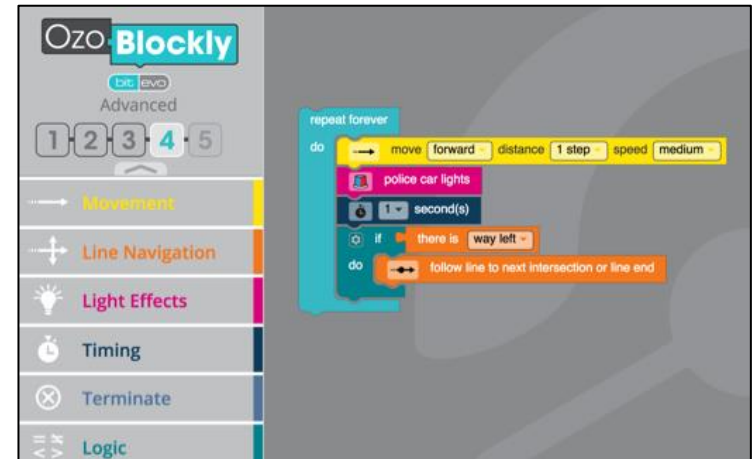
- The ProBot is the 'big brother' of the Bee-Bot
- More precision available than Bee-Bot, uses Logo programming language
- Programs can include procedures
- Touch, sound and light sensors
- Compatible with K'NEX



Digital devices

Ozobot

- Ozobot Bit can be programmed using text as in four colours
- Moves on to block based coding through OzoBlockly
- iOS and Android apps, web based programming

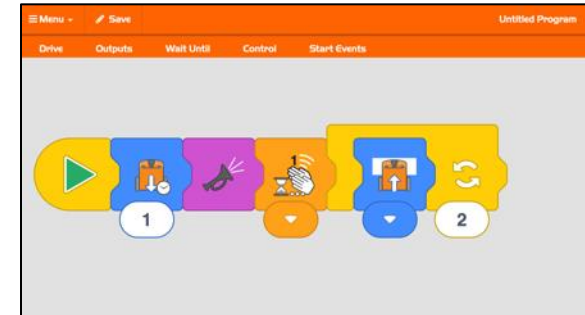
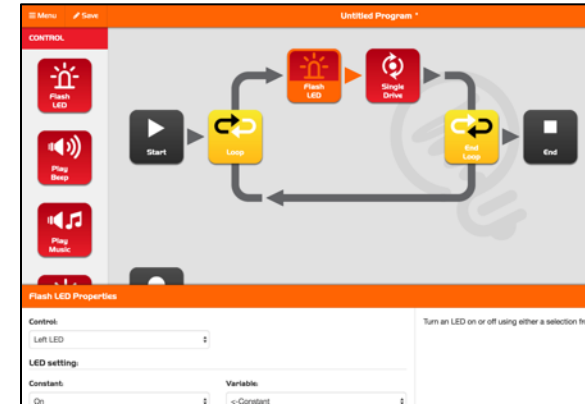
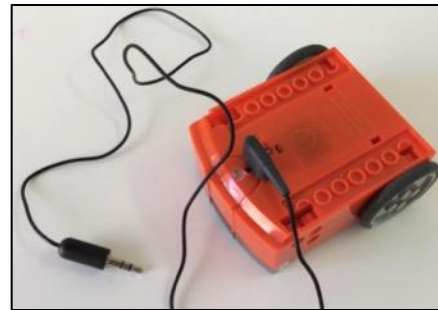


Digital devices

Edison

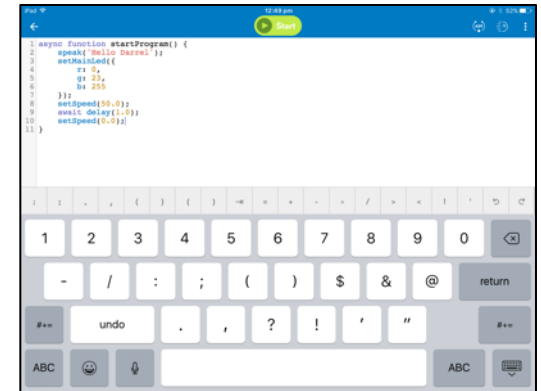
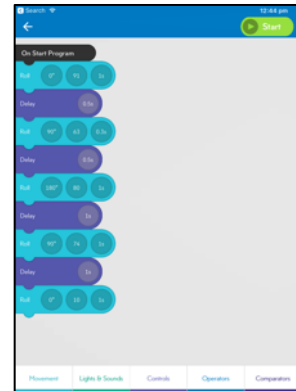
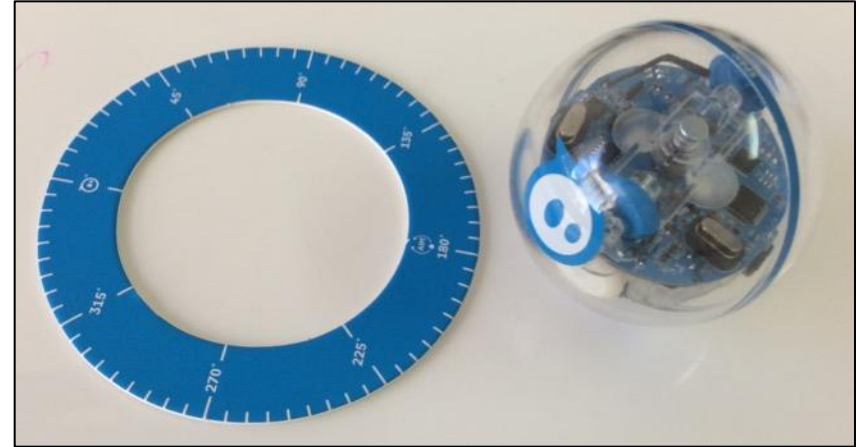
A robot that can be programmed using:

- Barcodes
- Computers / block language (online Edware & Edblocks App)
- Text based online - EdPy app
- Can play music
- Can be programmed to avoid obstacles and follow lines
- Compatible with Lego

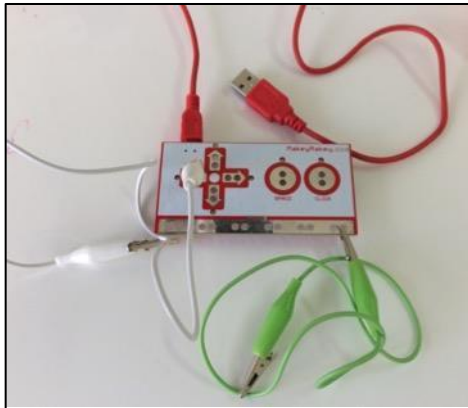
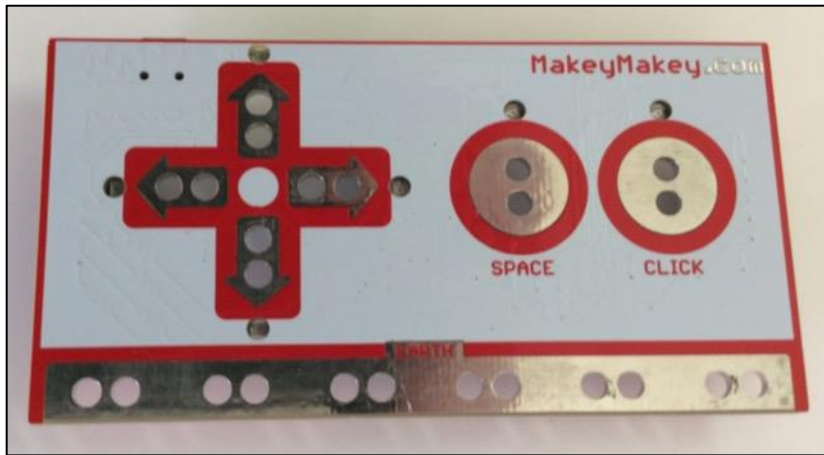


Sphero Sprk+

- Paired via Bluetooth to the Sphero Edu app
- Device can be programmed by 'drawing' on screen
- Can use block based programming
- Output includes movement, LEDs and sound (played through tablet/smartphone)
- Can be extended through text based programming (JavaScript)
- Has a good community for lesson ideas



Makey Makey

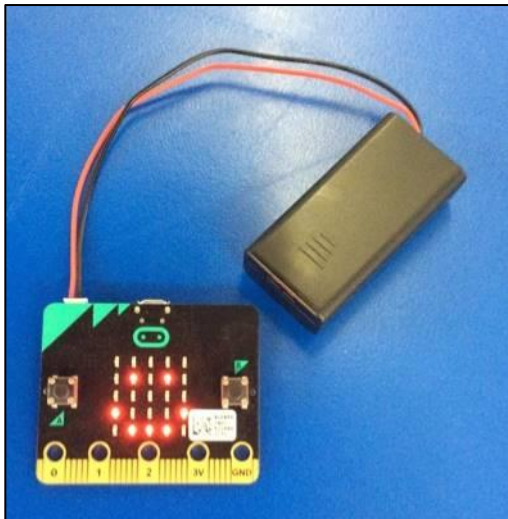


- Acts as an extension for a keyboard and mouse
- Other keys are accessible on reverse of board, as well as mouse movements
- Peripheral input, not processing
- Can be used in creative ways e.g., Game controller, reaction times, sound machine, 'smart bins', interactive displays

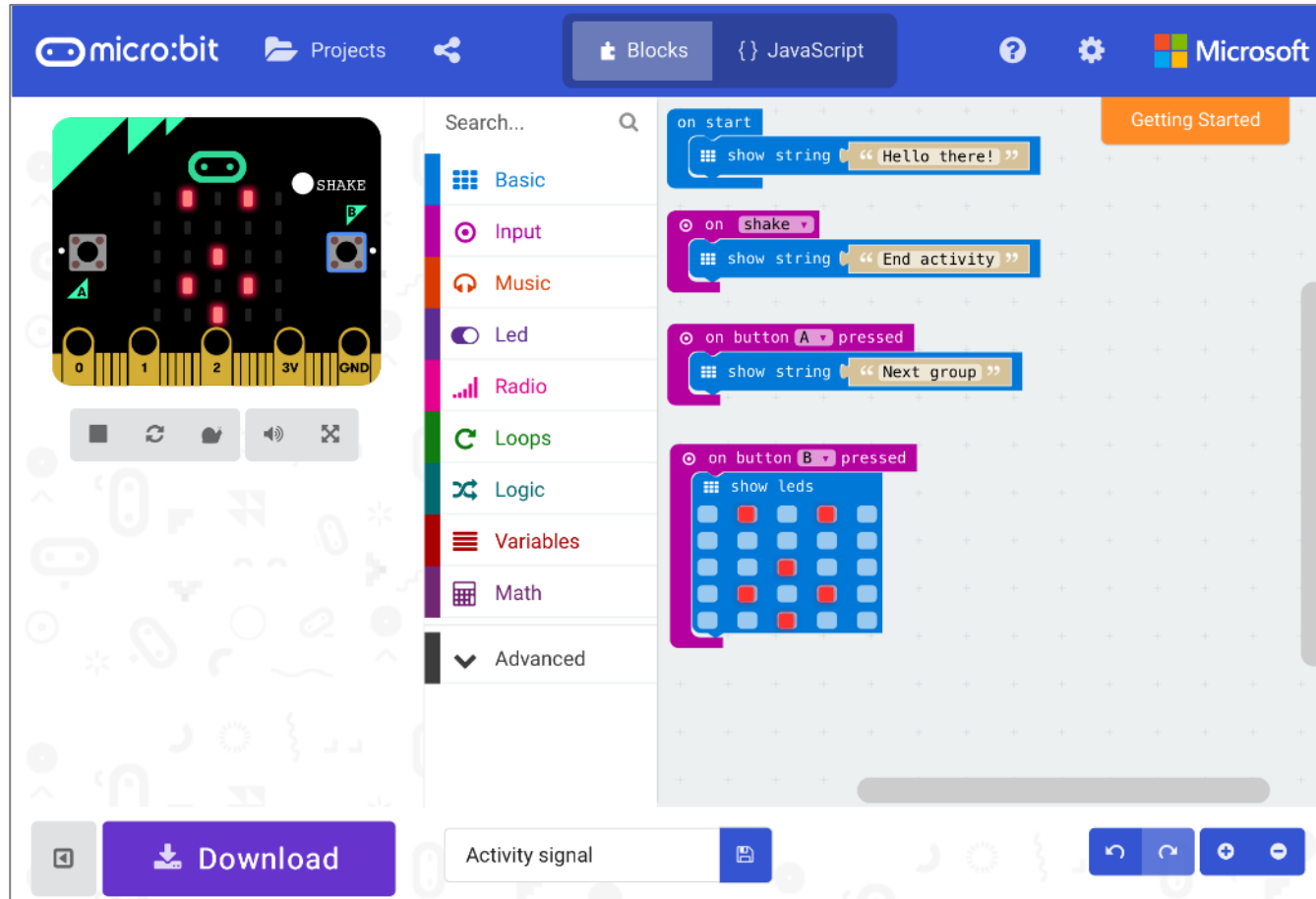
Digital devices

BBC micro:bit

- 5 by 5 grid of LEDs
- Two input buttons
- Bluetooth, compass & accelerometer
- 3 input pins and power out
- Block based coding or JavaScript
- Connects via micro USB



Coding the BBC micro:bit



Coding environment and simulation online at <https://makecode.microbit.org>

Digital devices

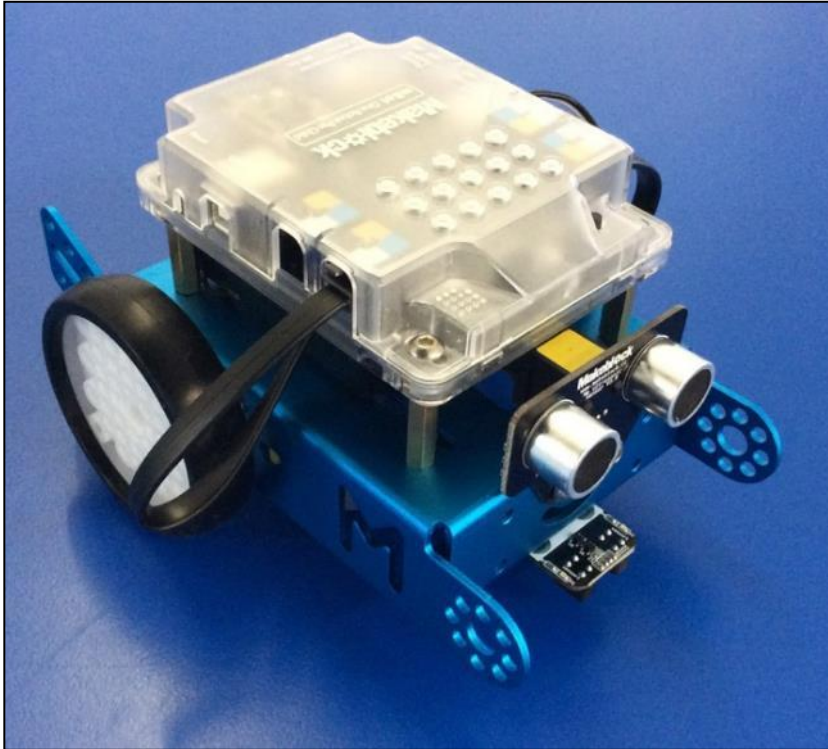
Dash & Dot

- Block based programming via tablet over Bluetooth
- Microphone, speakers, motors, lights, distance sensor
- Has a personality! (Preprogrammed behaviours and speech)
- Many iOS and Android apps



A screenshot of the Wonder Workshop website. The header includes the logo and navigation links: CleverBots, Robotics Competition, Education, Store, Sign In, and Buy Now. Below the header, there are tabs for Register Dash, Apps, Accessories, What's Included, and Specs. The main content area features a blue background with the headline "There are no limits to what your child can do with Dash" and a sub-headline "There's a Dash app for every age group, learning level, and play style." Below this, there are four app cards: Wonder (For ages 6+, picture-based coding), Blockly (For ages 6+, advanced concepts of coding), Path (For ages 6+, fundamentals of robotics), and Xylo (For ages 6+, musical performer). Each card includes a small icon of the Dash robot and a brief description of the app's features.

mBot

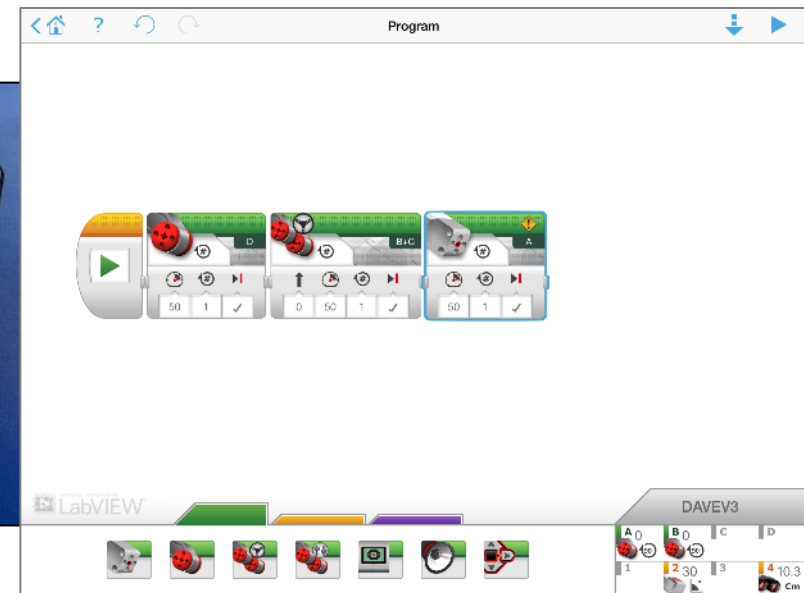


- Based on open source Arduino
- Block based programming through mBlock and MakeBlock apps
- Program on PC, Mac, iOS, Android and Chromebooks
- Line detection, obstacle avoidance, light sensor, lights, basic sound output, IR remote
- Connect via Bluetooth, USB or 2.4 Ghz with USB dongle
- Build your own!
- Can extend with other sensors and motors
- Compatible with Lego

Lego Mindstorms NXT and EV3



- Program on PC, Mac, iOS, Android and Chromebooks
- Block based programming
- Compatible with other Lego kits
- Many sensors
- Motors have rotational sensors

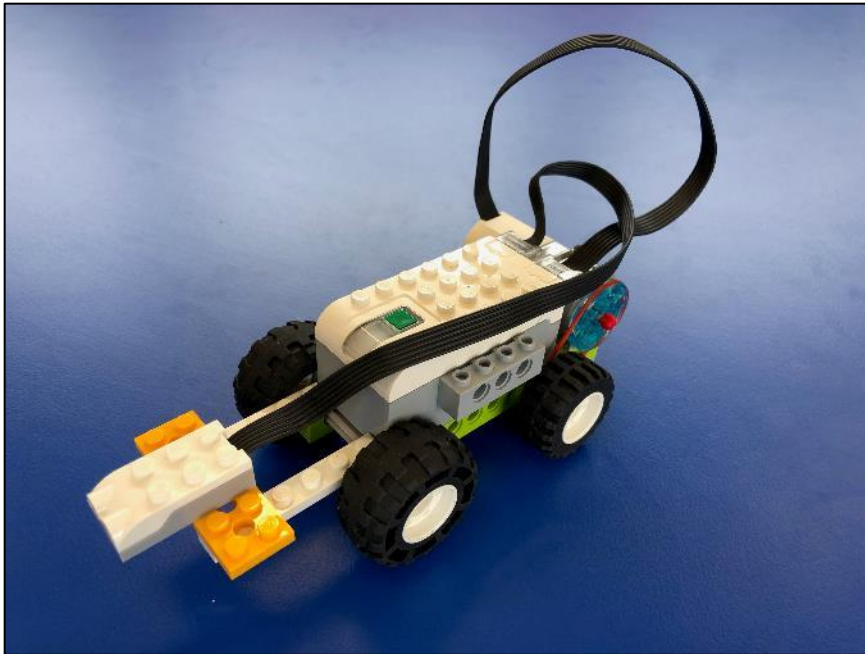


Hummingbird Duo Kit



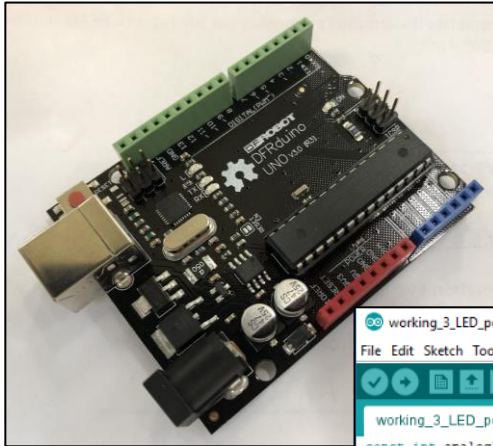
- Create and program robots built from electronic components and craft materials
- Based on open source Arduino
- Block based programming through SNAP! and Scratch on PC and Mac. Also BirdBlox on iOS and Android (with additional Bluetooth dongle).
- A wide array of sensors – light, temperature, distance, rotation

Lego WeDo



- Create objects and vehicles that are modifiable
- Interactive through light sensor, tilt sensor and a motor (although only two slots on 'smart brick')
- Block based programming via Lego WeDo app (iOS)
- Also programmable through Scratch

Arduino micro-controller boards



```
working_3_LED_potentiometer | Arduino 1.8.3
File Edit Sketch Tools Help
working_3_LED_potentiometer
const int analogInPin = A0; // Analog input pin that the potentiometer
const int analogOutPin1 = 9; // Analog output pin that the LED is
const int analogOutPin2 = 8; // Second output pin
const int analogOutPin3 = 7; // Other output pin

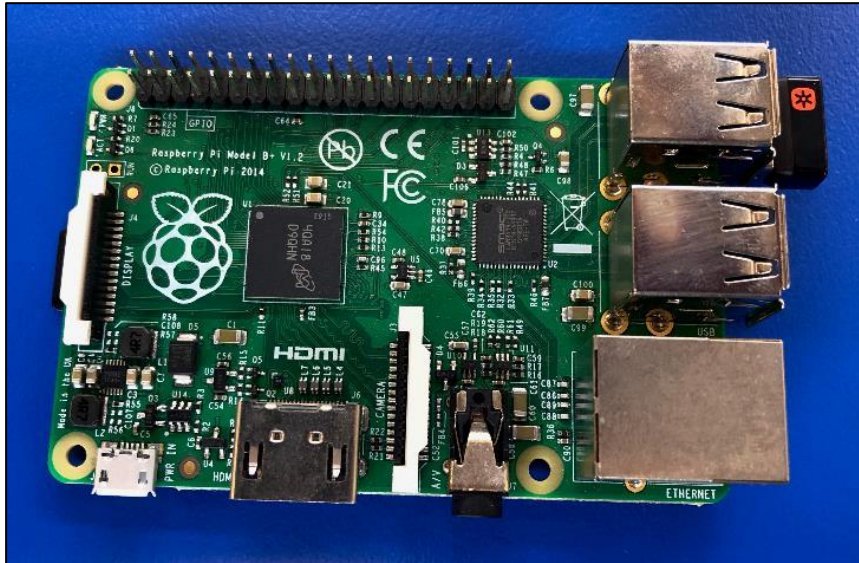
int sensorValue = 0; // value read from the potentiometer
int outputValue = 0; // value output to the analog pins (analogOutPin1)

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 99);
}
```

- Programmed using a C/C++ programming environment downloaded from <http://www.arduino.cc>
- Micro-controller that can take input, process data and produce output
- Connect to a range of sensors
- Many different models with capabilities such as Wi-fi and Bluetooth
- GPIO pins

Raspberry Pi



- Single board computer, usually runs on versions of Linux operating system
- HDMI and USB ports
- GPIO pins
- Version 3 and Zero-W have wireless and Bluetooth in-built
- Programmable via Scratch and Python (included in many basic software distributions)
- Requires peripherals for input and output (monitor, keyboard, mouse, etc)

Digital Devices in the curriculum

Where does it fit?

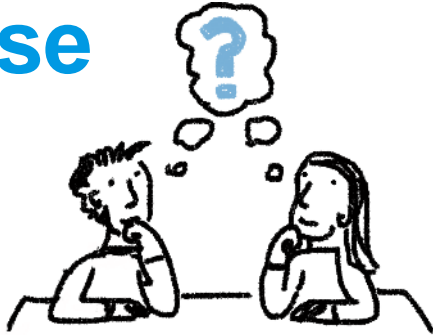
	Input/output	Branching	User input	Iteration	Functions	Modules
Bee-Bot	✓		✓			
ProBot	✓	✓	✓	✓	✓	
OzoBot	✓	✓	✓	✓	✓	
Makey Makey	✓					
BBC micro:bit	✓	✓	✓	✓	✓	✓
mBot	✓	✓	✓	✓	✓	✓
Lego WeDo/ NXT/EV3	✓	✓	✓	✓		

Where does it fit?

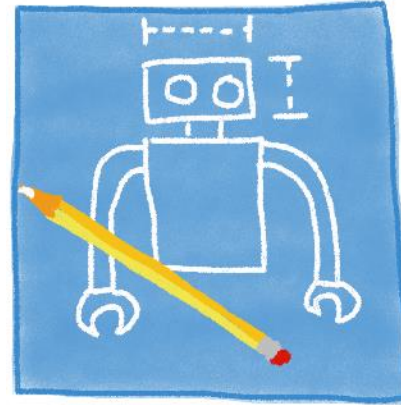
	Input/output	Branching	User input	Iteration	Functions	Modules
Sphero	✓	✓	✓	✓	✓	✓
Edison	✓	✓	✓	✓	✓	✓
Arduino	✓	✓	✓	✓	✓	✓
Raspberry Pi	✓	✓	✓	✓	✓	✓
Dash & Dot	✓	✓	✓	✓		
Hummingbird	✓	✓	✓	✓	✓	✓

Using digital devices for learning

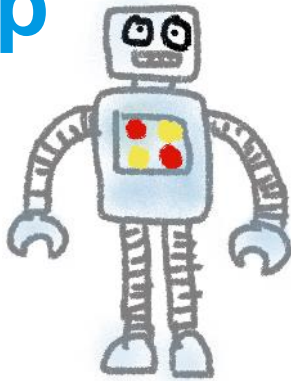
Analyse



Design



Develop



Evaluate



Using digital devices for learning

- Collaborate
 - Cultivate collective expertise
 - Problem solving and 'beta testing'
 - Simulate code by writing, flowcharting, walking through
- Audience
 - Who sees/uses/evaluates finished solution?
- Mix of plugged and unplugged learning
 - Transfer across context
 - Same concepts on multiple platforms/environments/devices
- Cross curriculum
 - Skills that can be used to demonstrate understanding in other curriculum areas

Using digital devices for learning

- Design your assessment at the planning stage
 - What skills will students need?
 - How will those skills be demonstrated?
 - Solo/partner/group work
- Attempt your tasks first!
 - Find the tricky points where explicit teaching is necessary
 - 'Just in time' workshops
 - Target your time and support materials (from anchor charts to support websites or FAQs)
- Celebrate successes
- Celebrate failure!