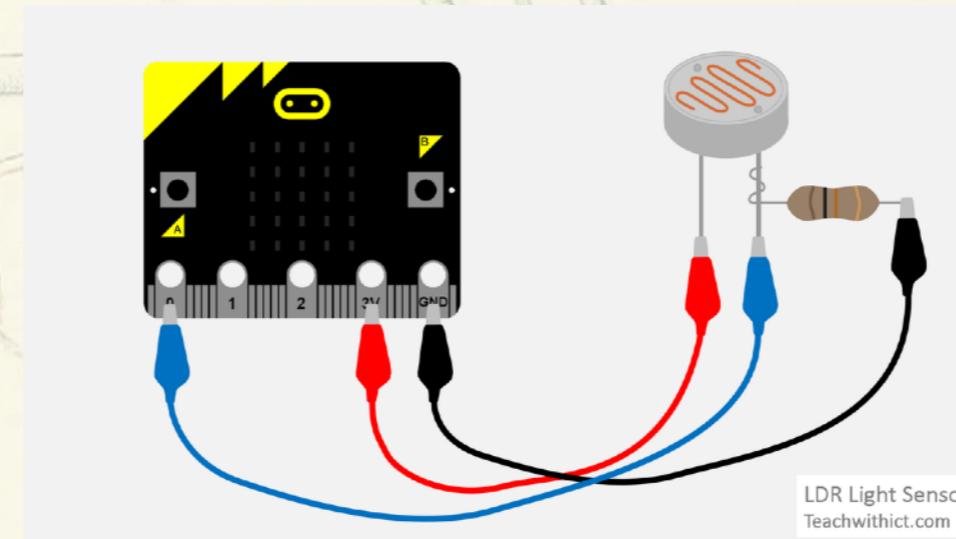


Corso Robotica educativa: Tinkering e elettronica educativa (con Micro:bit)

Lezione 1, Micro:bit



Link al materiale del corso

[https://github.com/conradis/Microbit Tinkering/](https://github.com/conradis/Microbit_Tinkering/)



Presentazione del corso

Lezione 1

- Micro:bit, cosa è, hardware ed elettronica
- Kit micro:bit
- Programmare micro:bit : cosa significa? come?
- Makecode, Python.
- Programmare con Makecode.

Presentazione del corso

Lezione 2

- Tinkering e didattica
- Robotica educativa ed elettronica nella scuola secondaria di primo grado
- Attività pratiche ed esempi didattici

Presentazione del corso

Lezione 3

- Tinkering e didattica
- Robotica educativa ed elettronica nella scuola primaria
- Attività pratiche ed esempi didattici

Finalità del corso

Il corso ha l'obiettivo di inserire in tinkering nella didattica curricolare come forma di apprendimento informale in cui si “impara facendo”.

Il materiale didattico proposto si configura come metodo e strumento attraverso cui il docente incoraggerà a sperimentare, a stimolare l'attitudine alla risoluzione dei problemi, a insegnare a lavorare in gruppo e a collaborare per il raggiungimento di un obiettivo, riuscendo a far accostare bambini e adolescenti a discipline come l'arte, la scienza e la tecnologia attraverso la manipolazione dei materiali.

Il Corso intende promuovere una didattica inclusiva e collaborativa, che accompagni i docenti verso un nuovo modo di fare e recepire la didattica.

L'uso di modelli pedagogici innovativi come strumenti per potenziare la didattica tradizionale, da un lato rafforza l'interesse dei docenti per l'aggiornamento delle proprie competenze, dall'altro favorisce la personalizzazione dei percorsi di apprendimento e trasforma gli studenti in attori consapevoli e attivi del loro percorso di crescita.

Obiettivi

I corso si pone come obiettivo principale quello di inserire il Tinkering nella didattica curricolare presentando una serie di attività pratiche per:

- favorire l'utilizzo di nuovi approcci e modelli di insegnamento/apprendimento capaci di mettere gli alunni al centro del processo formativo;
- favorire un ampliamento dei percorsi curriculari per lo sviluppo ed il rinforzo delle competenze;
- favorire l'apprendimento interdisciplinare e multidisciplinare attraverso modalità didattiche mediate dal tinkering;
- incentivare la produzione di learning objects, o semplicemente di materiali didattici, da condividere all'interno della scuola;
- guidare i docenti nell'adozione del tinkering a scuola come strumento didattico multidisciplinare a disposizione degli insegnanti.

Il pensiero computazionale

L'intero corso e in particolare le attività proposte sono centrate sul concetto di "pensiero computazionale". Di cosa si tratta?

Nel 2006 Jeannette Wing, direttrice del Dipartimento di informatica della Carnegie Mellon University, riuscì a formulare una definizione capace di mettere d'accordo molti studiosi:

"Il pensiero computazionale è un processo di formulazione di problemi e di soluzioni in una forma che sia eseguibile da un agente che processi informazioni."

La stessa Wing ha inoltre messo a fuoco alcune caratteristiche del Computational Thinking: esso non consiste semplicemente nel saper programmare, ma nel pensare a diversi livelli di astrazione; è un'abilità fondamentale per tutti, che dovrebbe diventare la quarta abilità di base oltre al saper "leggere, scrivere e fare di conto".

Il pensiero computazionale

Il pensiero computazionale è un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici, pianificando una strategia; abitua al rigore e quindi rende possibili gli atti creativi. Permette di interagire con persone e strumenti, di fruire delle potenzialità delle macchine quali oggetti capaci di compensare le lentezze o l'imprecisione dell'uomo, se ben programmati.

Come tutte le scienze, ha i suoi fondamenti formali nel linguaggio matematico e ha a che fare con oggetti del mondo reale: il pensiero computazionale “*Complements and combines mathematical and engineering thinking*” (Wing).

Il pensiero computazionale attiene al mondo delle idee e delle strategie, è per tutti, in ogni luogo: può essere applicato a qualunque situazione ed essere appannaggio non solo degli informatici.

Guardare il mondo con occhio matematico

Sviluppare la capacità di creare modelli matematici/logici di ciò che abbiamo attorno.

*“Se la gente non crede che la matematica sia semplice,
è soltanto perché non si rende conto di quanto
complicata sia la vita.”*

(John von Neumann)

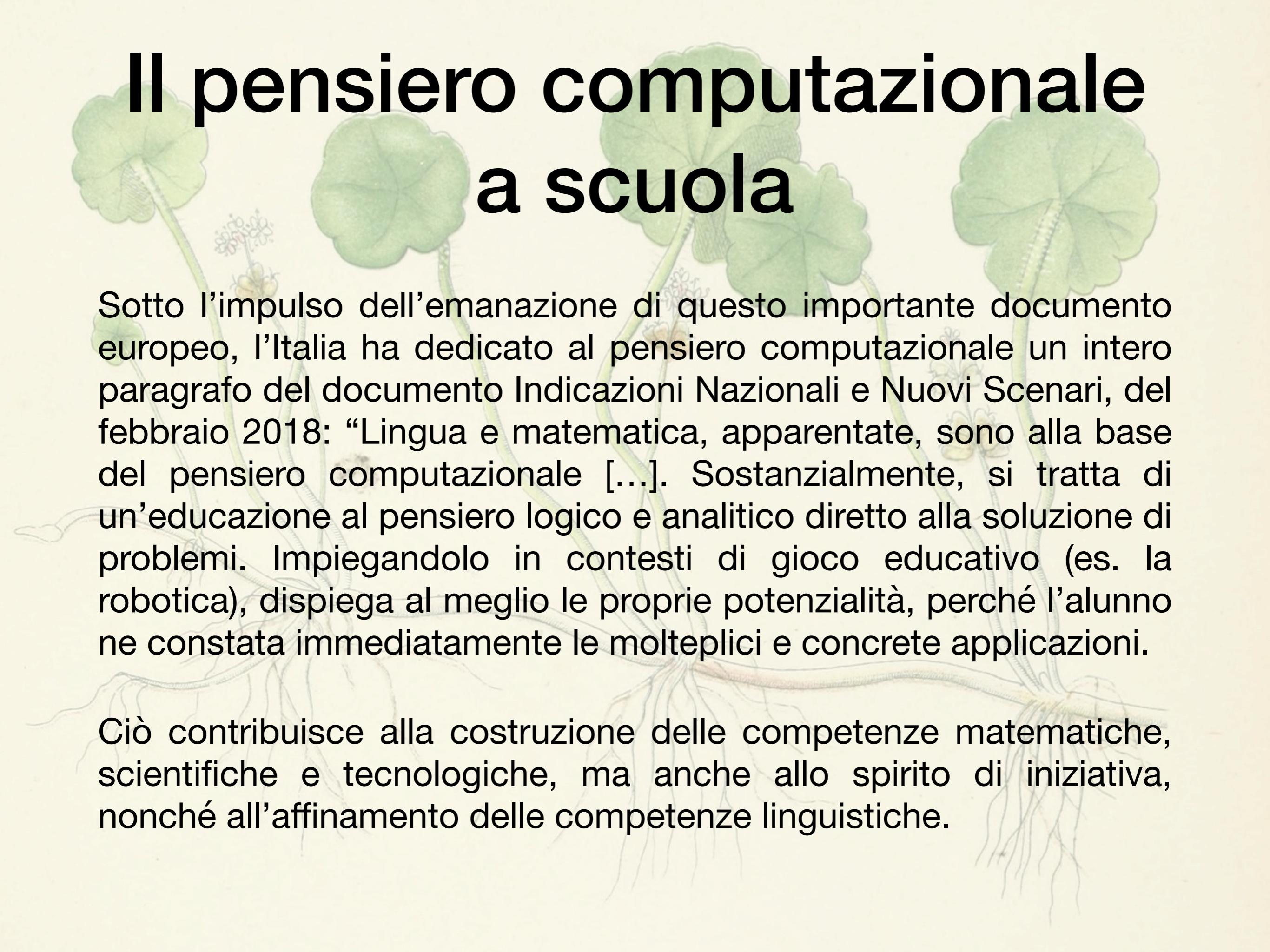
Il pensiero computazionale a scuola

La legge 107 del 13 luglio 2015 ha collocato il pensiero computazionale tra gli strumenti culturali per la cittadinanza, cioè tra quelle abilità e competenze che realizzano la piena cittadinanza.

In ambito europeo, la Commissione Europea ha emanato il Digital Education Action Plan (Brussels, 2018), che ha stabilito alcune priorità nell'ambito dello sviluppo delle competenze digitali degli studenti e dei cittadini europei.

Tra queste priorità, la seconda riguarda le competenze computazionali e di programmazione: “**L’acquisizione delle competenze digitali deve iniziare in giovane età e continuare per tutta la vita. Ciò è possibile nell’ambito dei programmi di studio o a livello extra-scolastico. I giovani europei sono accaniti utenti di Internet, app e giochi, ma devono anche conoscere meglio le strutture soggiacenti e gli algoritmi di base e diventare creatori e leader digitali”.**

Il pensiero computazionale a scuola



Sotto l'impulso dell'emanaione di questo importante documento europeo, l'Italia ha dedicato al pensiero computazionale un intero paragrafo del documento Indicazioni Nazionali e Nuovi Scenari, del febbraio 2018: “Lingua e matematica, apparentate, sono alla base del pensiero computazionale [...]. Sostanzialmente, si tratta di un’educazione al pensiero logico e analitico diretto alla soluzione di problemi. Impiegandolo in contesti di gioco educativo (es. la robotica), dispiega al meglio le proprie potenzialità, perché l’alunno ne constata immediatamente le molteplici e concrete applicazioni.

Ciò contribuisce alla costruzione delle competenze matematiche, scientifiche e tecnologiche, ma anche allo spirito di iniziativa, nonché all'affinamento delle competenze linguistiche.

Il pensiero computazionale nella società

Nei contesti attuali, in cui la tecnologia dell'informazione è così pervasiva, la padronanza del coding e del pensiero computazionale possono aiutare le persone a governare le macchine e a comprenderne meglio il funzionamento, senza esserne invece dominati e asserviti in modo acritico.

A proposito di asservimento...

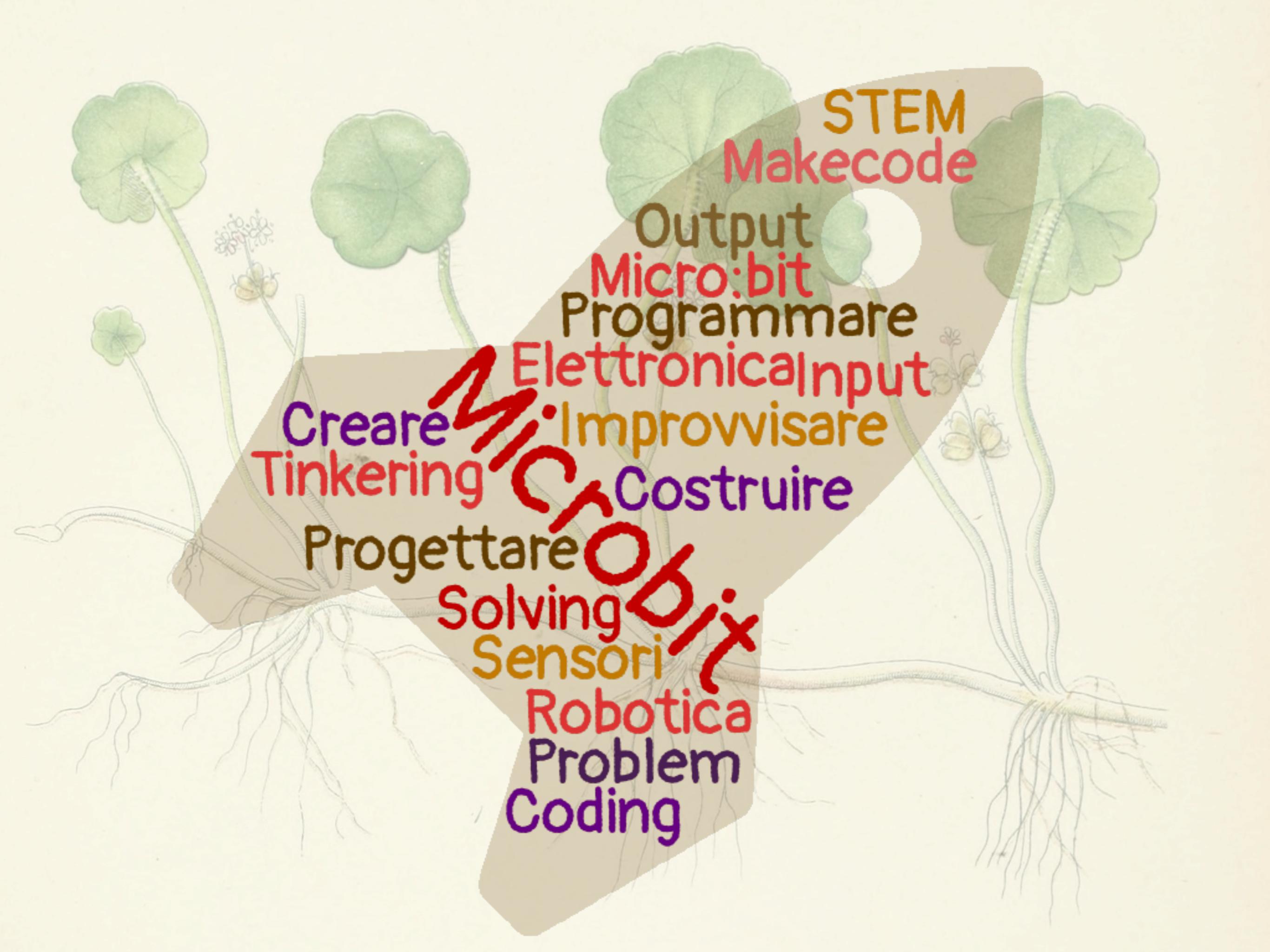
Gli esempi sarebbero numerosi...



Coltivare...

... il pensiero computazionale e applicarlo in modo interdisciplinare perché costituisce una sorta di fertilizzante che prepara il terreno sia per l'uso consapevole della tecnologia sia per comprendere gli aspetti logici e la struttura profonda delle attività che si svolgono.

Il pensiero computazionale non ha bisogno della tecnologia, viene prima della tecnologia: è un'abilità trasversale, un processo di problem solving utile in qualunque contesto che poggia i suoi fondamenti sulla matematica e sulla logica.



STEM

Makecode

Output

Micro:bit
Programmare

Elettronicalnput

Create
Tinkering

Improvvisare

Costruire

Progettare

Solving

Sensori

Robotica

Problem

Coding

Il tinkering

Che cos'è il Tinkering?

Tinkering è un termine inglese che vuol dire letteralmente “armeggiare, adoperarsi, darsi da fare”.

Il Tinkering viene oramai considerato, negli ambienti educativi a livello internazionale, un approccio innovativo per l'educazione alle STEM, ed è menzionato nel PIANO NAZIONALE SCUOLA DIGITALE come uno strumento importante per lo sviluppo delle competenze del 21° secolo e per l'educazione alle STEM.

Si parla di tinkering come di una forma di apprendimento informale in cui si impara facendo. L'alunno è incoraggiato a sperimentare, stimolando in lui l'attitudine alla risoluzione dei problemi.

Le attività vengono lanciate principalmente sempre sotto forma di gioco o sfida.

BBC Micro:bit



BBC Micro:bit

Microbit (BBC Micro:bit) è un microcontrollore open source basato su ARM, progettato dalla BBC per la computer education nel Regno Unito. La sua diffusione è cominciata dal Febbraio 2016 e da allora ne sono stati venduti oltre 2 milioni!

<https://video.microbit.org/microbit-video-education-04052017-nomusic.mp4#t=2>

Il dispositivo è poco più piccolo di una carta di credito e ha un processore ARM Cortex-M0, un accelerometro, un magnetometro, connettività USB e Bluetooth, un display 5 x 5 fatto da 25 LED, due pulsanti programmabili e può essere alimentato tramite USB oppure tramite batterie. Gli input e gli output avvengono attraverso 5 contatti ad anello e 25 pin ulteriori.

Micro:bit e ScuolaDigitale TIM

<https://youtu.be/Q48-xX6SDIQ>



Simon Peyton Jones - Revolution in computing education at school: opportunity & challenge

- <https://youtu.be/X-Albtus9gs>

Computing in action

'M' realised she could replace the repeated steps by this for loop

```
function main ()  
    turtle → forward(100)  
    turtle → right turn(90)  
    repeat 4 times  
        turtle → forward(100)  
        turtle → right turn(90)  
    end repeat  
end function
```

```
function main ()  
    for 0 ≤ i < 4 do  
        turtle → forward(100)  
        turtle → right turn(90)  
    end for  
end function
```

The revolution in computing education at school: opportunity and challenge

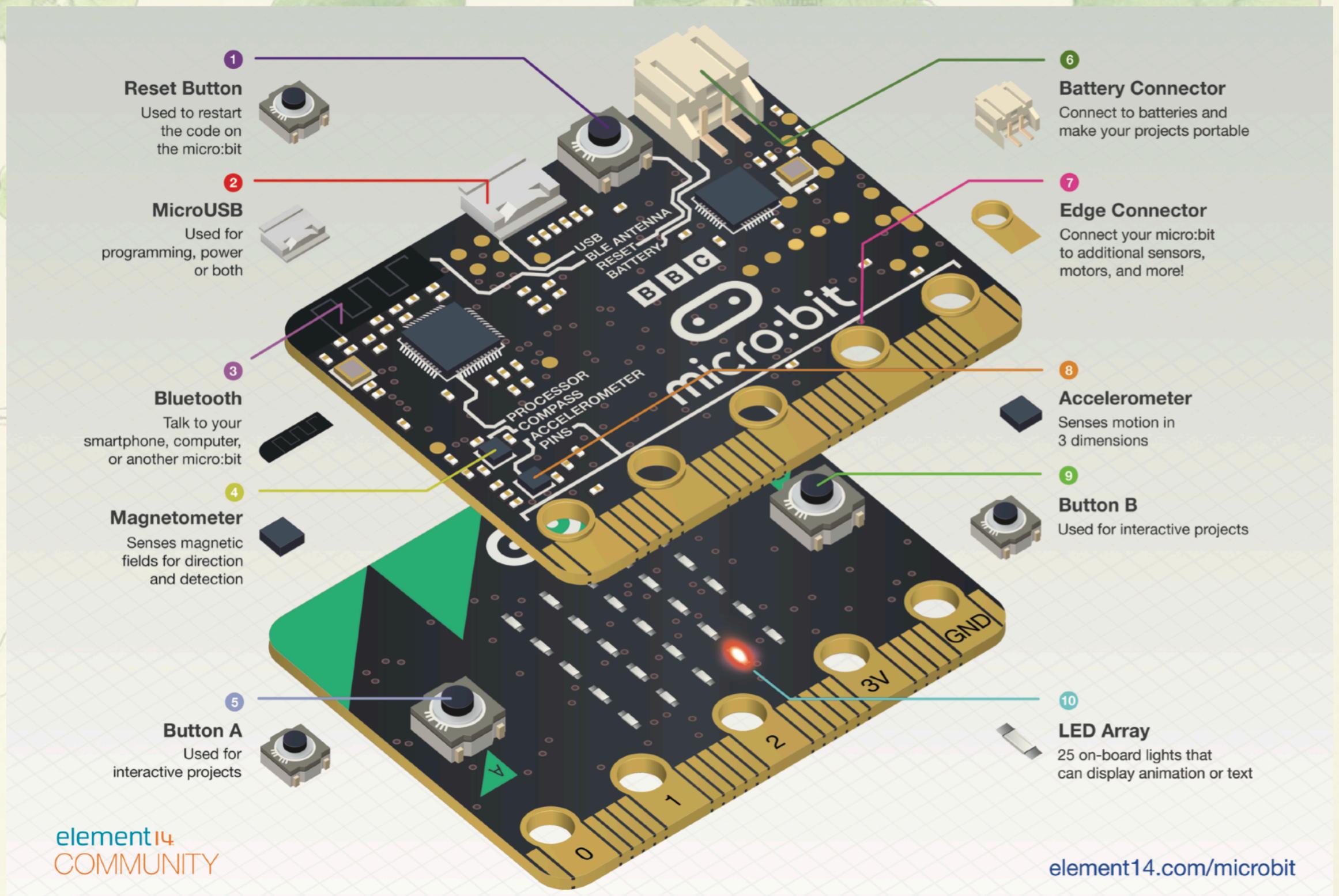
Simon Peyton Jones

46:01 / 55:26

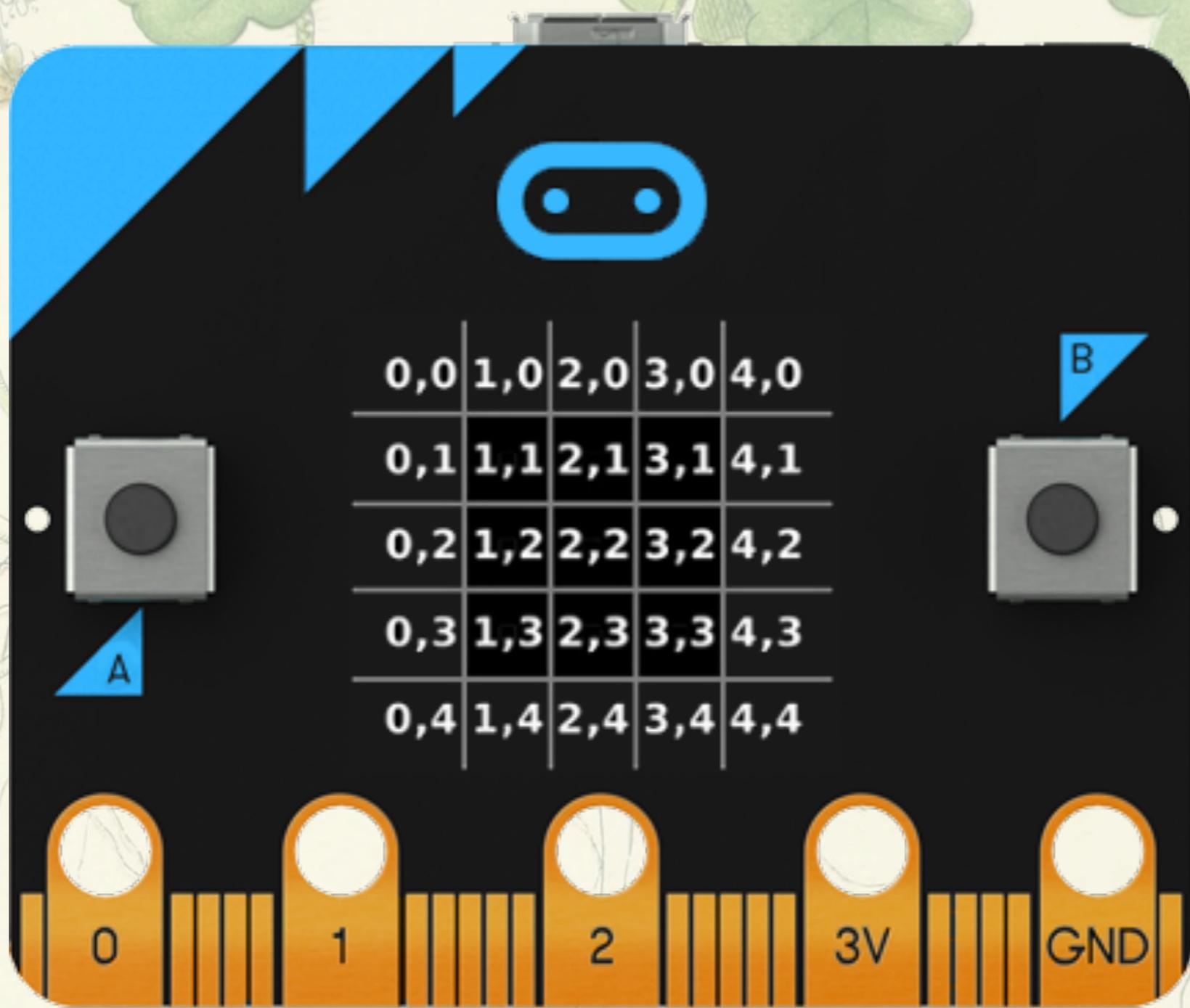
BBC Micro:bit

- The physical board measures 43 mm × 52 mm and, in its first production run,[10] included:
- Nordic nRF51822 – 16 MHz 32-bit ARM Cortex-M0 microcontroller, 256 KB flash memory, 16 KB static RAM, 2.4 GHz Bluetooth low energy wireless networking. The ARM core has the capability to switch between 16 MHz or 32.768 kHz.[2][3][11][12][13][14]
- NXP/Freescale KL26Z – 48 MHz ARM Cortex-M0+ core microcontroller, that includes a full-speed USB 2.0 On-The-Go (OTG) controller, used as a communication interface between USB and main Nordic microcontroller. This device also performs the voltage regulation from the USB supply (4.5-5.25 V) down to the nominal 3.3 volts used by the rest of the PCB. When running on batteries this regulator is not used.
- NXP/Freescale MMA8652 – 3-axis accelerometer sensor via I²C-bus.[15]
- NXP/Freescale MAG3110 – 3-axis magnetometer sensor via I²C-bus (to act as a compass and metal detector).[15]
- MicroUSB connector, battery connector, 25-pin edge connector.[2][13][15][16]
- Display consisting of 25 LEDs in a 5×5 array.[13]
- Three tactile pushbuttons (two for applications, one for reset).[17]
- I/O includes three ring connectors (plus one power one ground) which accept crocodile clips or 4 mm banana plugs[17] as well as a 25-pin edge connector with two or three PWM outputs, six to 17 GPIO pins (depending on configuration), six analog inputs, serial I/O, SPI, and I²C.[15] Unlike early prototypes, which had an integral battery, an external battery pack (AAA batteries) can be used to power the device as a standalone or wearable product.[3][9][13] Health and safety concerns, as well as cost, were given as reasons for the removal of the button battery from early designs.[18]
- The available hardware design documentation consist of only the schematic and BOM distributed under the Creative Commons By Attribution license, no PCB layout is available.[19][20] The compatible reference design by Micro:bit Educational Foundation, however, is fully documented.

BBC Micro:bit



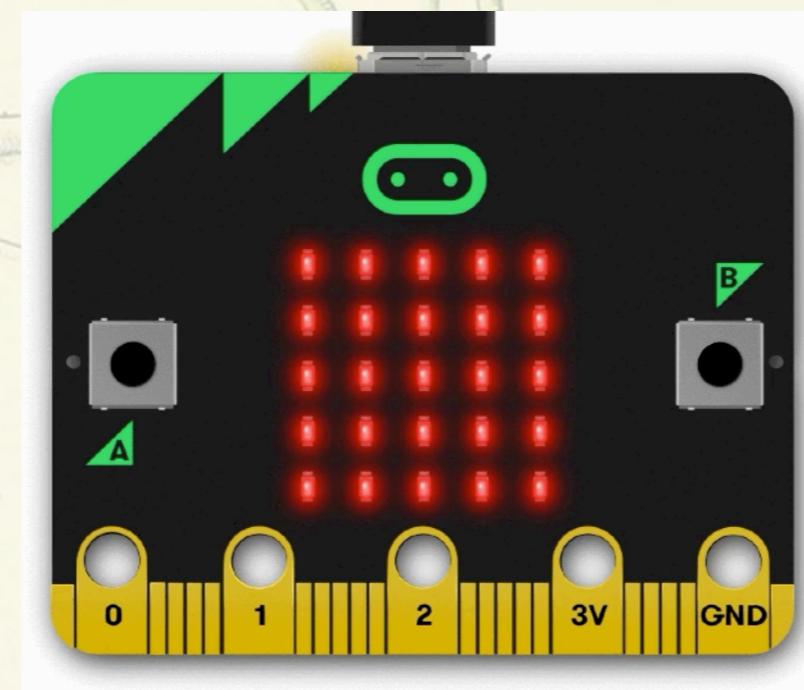
II display a LED



<https://microbit.org>

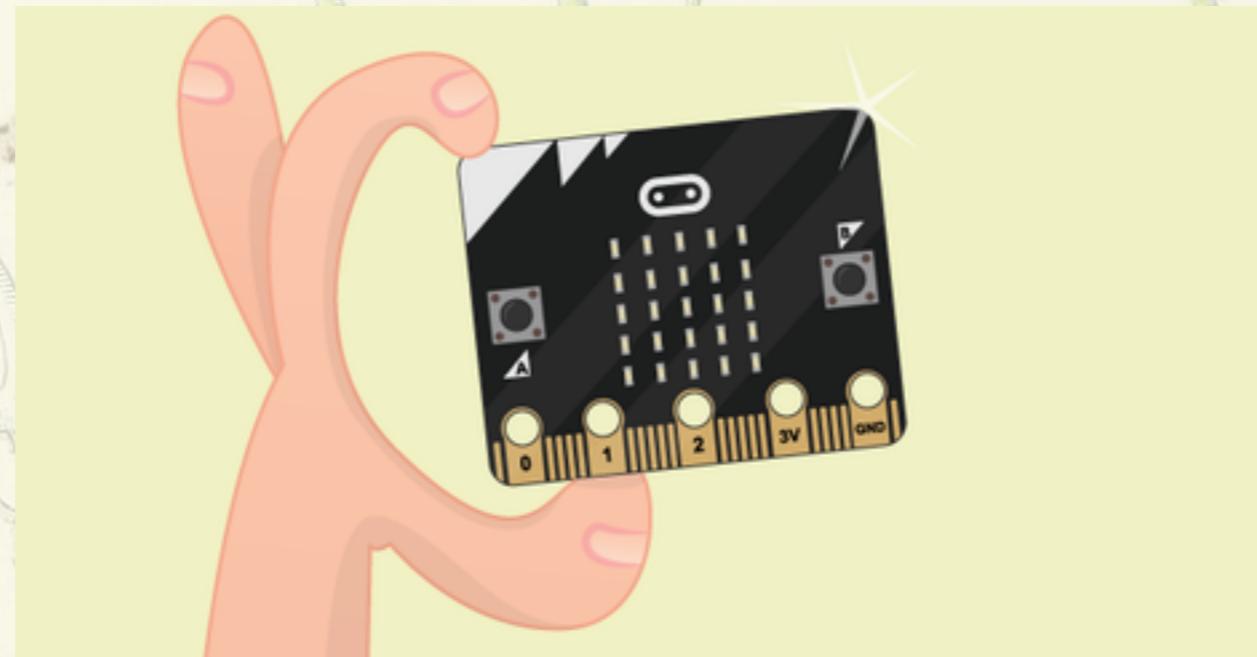
Diamo un'occhiata al sito ufficiale, che sarà il riferimento primario per qualunque attività didattica si voglia implementare:

<https://microbit.org>



Come usare micro:bit

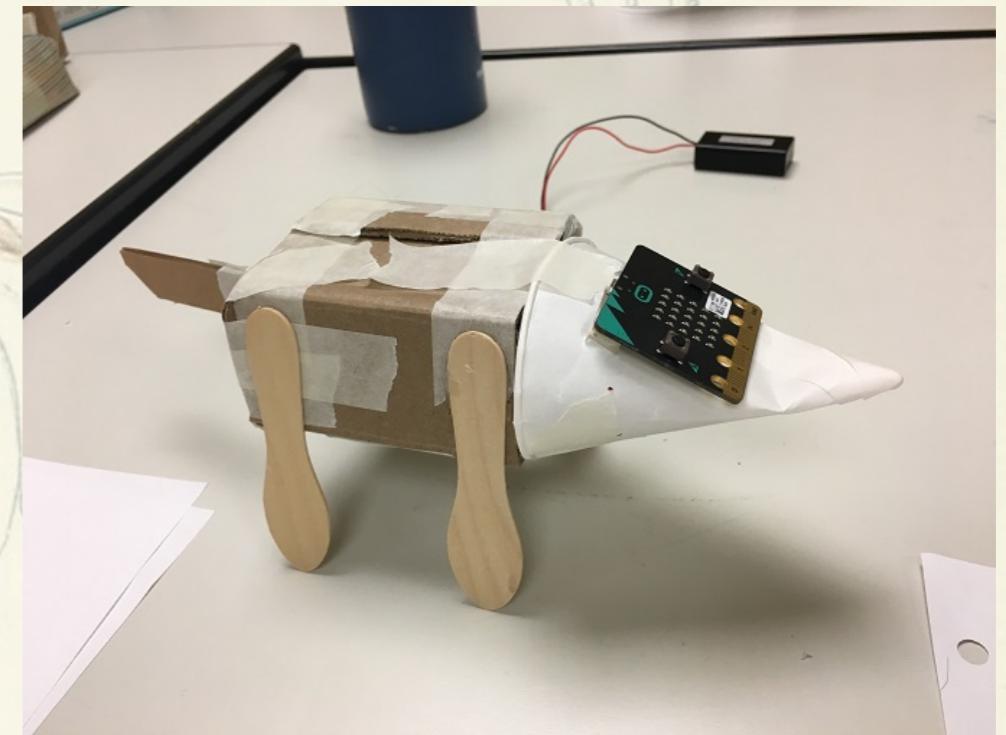
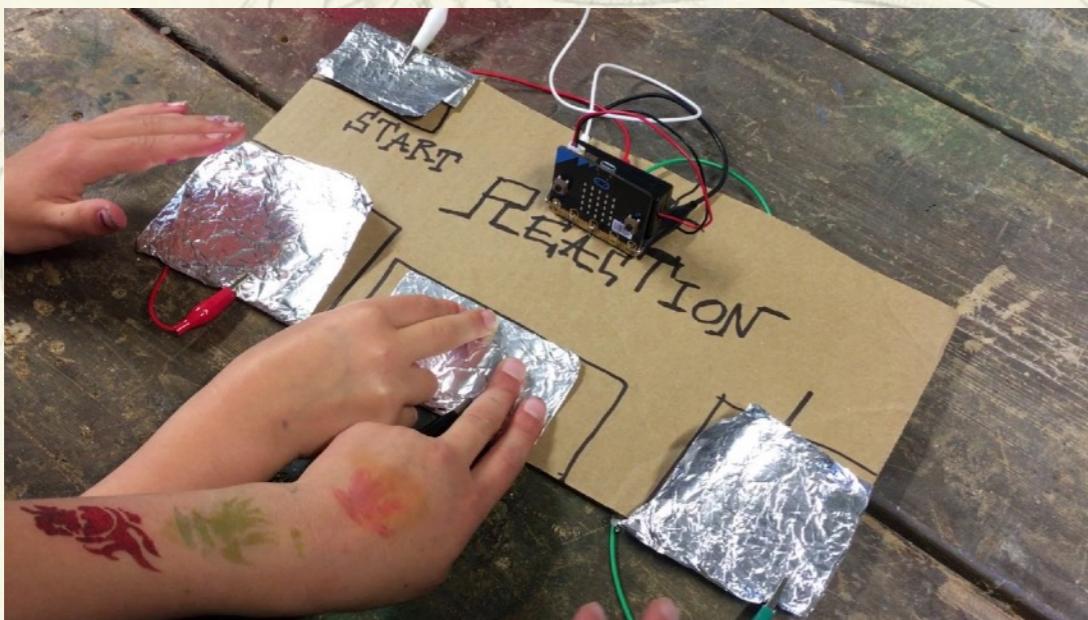
Micro:bit diventa il cuore (e anche il cervello) dei nostri esperimenti, dei nostri progetti di robotica, di elettronica e di informatica oppure delle nostre attività STEM.



Ma come? Vedremo molti dettagli durante il corso, ma per il momento possiamo semplificare ogni progetto micro:bit come costituito da due parti:

1) Hardware

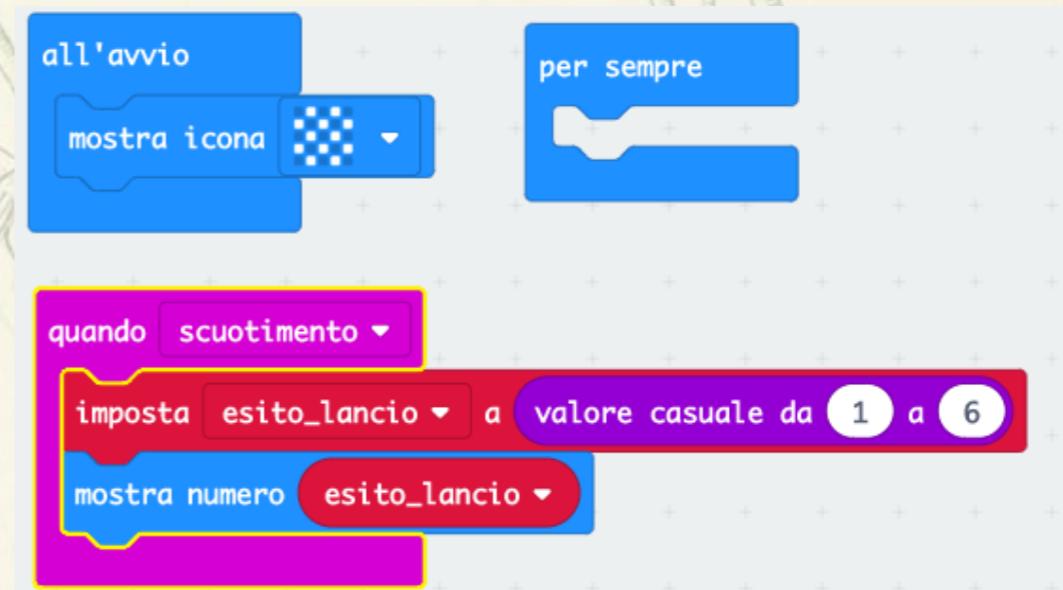
- I dispositivi elettrici o elettronici collegati ed il micro:bit
- Le parti in plastica, cartone, metallo, Lego.
- Motori o sensori esterni.
- Tutto quello che serve per far funzionare il nostro esperimento e per renderlo carino!



2) Software

- Il codice che permette di **programmare** micro:bit affinché si comporti nel modo che abbiamo previsto.
- La colla che unisce tutte le parti del nostro esperimento.
- Il cervello del nostro robot.

```
from microbit import *\n\nwhile True:\n    reading = accelerometer.get_x()\n    if reading > 20:\n        display.show("R")\n    elif reading < -20:\n        display.show("L")\n    else:\n        display.show("-")
```



Micro:bit nella didattica risultati: UK 2017

Studio commissionato dalla BBC dopo aver donato gratuitamente circa 1 milione di Micro:bit a

- tutti gli studenti di 7 anni in Inghilterra e Galles
- tutti gli studenti di 8 anni in Irlanda del Nord
- tutti gli studenti di 11 anni in Scozia

Lo studio ha evidenziato che:

- Il 90% degli studenti ritiene che micro:bit abbia loro mostrato che chiunque possa fare coding.
- L' 86% degli studenti ha affermato che micro:bit ha reso la didattica più interessante
- Il 70% delle ragazze ha detto che, dopo aver usato micro:bit, sceglierrebbe Informatica come materia di scuola.
- L'85% degli insegnanti ha affermato che micro:bit ha reso le materie di ICT/Computer Science più piacevoli per i loro studenti.
- Metà degli insegnanti che hanno usato micro:bit si sentono più confidenti nel loro ruolo, in particolare coloro che non erano molto confidenti nel Coding e TIC

Micro:bit nella didattica risultati: Danimarca 2019

1447 scuola Danesi (su un totale di circa 1600) hanno siglato un progetto comune (ultra:bit). Il Teachers Resource Center ha fornito BBC Micro:bit a 64287 studenti.

Il Center for Evaluation and Development of Science Education (NEUC) ha valutato che

- Il 90% degli insegnanti ha trovato più facile il Coding dopo aver lavorato con micro:bit.
- Il 95% degli insegnanti ha rilevato che i loro studenti abbiano migliorato le loro skills di coding.

Il software, ovvero come programmare micro:bit

Micro:bit è programmabile in diversi modi e con vari linguaggi.

I linguaggi ufficiali sono:

- Microsoft Makecode (e Javascript)
- Micropython



Sono supportati numerosi altri linguaggi....

<https://github.com/carlosperate/awesome-microbit#-other-languages>

Il corso sarà focalizzato su Makecode, ma prima...

MicroPython

Python è uno dei linguaggi più popolari al mondo. Ogni giorno utilizziamo software scritto in Python is one of the world's most popular programming languages.

Alcune aziende che usano Python per diverse applicazioni: Google, NASA, Bank of America, Disney, CERN, YouTube, Mozilla, The Guardian - e la lista continua ricoprendo tutti i settori che vanno dalla finanza all'arte.

Per esempio ricordate l'annuncio della scoperta delle onde gravitazionali con l'esperimento LIGO? Gli strumenti di misura di quell'esperimento sono controllati con software scritto in Python.

Saper programmare in Python è una competenza di grande valore applicabile dall'informatica, all'arte o alla linguistica.

MicroPython

MicroPython è una implementazione agile e efficiente di Python 3 e include un subset della libreria standard di Python 3.

MicroPython è ottimizzato per eseguire su microcontrollori in sistemi embedded o con risorse limitate.

MicroPython su micro:bit contiene una speciale libreria **microbit** che permette di controllare la nostra schedina e tutte le sue funzionalità.

References: <https://microbit-micropython.readthedocs.io/en/latest/>

MicroPython Zen

MicroPython v1.9.2-34-gd64154c73 on 2017-09-01; micro:bit v1.0.1 with nRF51822
Type "help()" for more information.

>>> import this

The Zen of MicroPython, by Nicholas H. Tollervey

Code,
Hack it,
Less is more,
Keep it simple,
Small is beautiful,

Be brave! Break things! Learn and have fun!

Express yourself with MicroPython.

Happy hacking! :-)

Microsoft MakeCode

Microsoft MakeCode avvicina gli studenti all'informatica con progetti divertenti, risultati immediati ed editor di blocchi per studenti di livelli diversi.

Microsoft MakeCode è un framework che permette di creare software interattivo a coloro che non hanno mai programmato un calcolatore. Permette di fare coding, quindi creare ed eseguire programmi sia su micro:bit che su un micro:bit simulato.

<https://www.microsoft.com/it-it/makecode?rtc=1>

E ora?

You'll have most success if you explore, experiment and play. You can't break a BBC micro:bit by writing incorrect code. Just dive in!

Un principio da seguire: sbaglirai molte volte ed è un bene! I migliori sviluppatori di software imparano sbagliando: questo è il loro segreto che li ha resi così bravi.

Risorse online

- <https://microbit.org>
- <https://github.com/carlosperate/awesome-microbit>
- <https://microbit-micropython.readthedocs.io/en/latest/index.html>
- <https://www.teachwithict.com/physical-computing.html>

Si parte!

The image shows the Microsoft Scratch micro:bit interface. On the left, there is a preview of the micro:bit board with various pins and components labeled. A script is displayed on the right, consisting of the following blocks:

- An **on start** hat block with two **show string** blocks: one with "Hello!" and another with "0".
- An **on shake** hat block with a **clear screen** block.
- A **if** control block with a **Random** pick block set to 2. It has three branches:
 - A **show string** block with "YES".
 - An **else if** control block with a **Random** pick block set to 1. It has one branch:
 - A **show string** block with "NO".
 - An **else** control block with a **show string** block with "I DON'T KNOW".

MakeCode offline

<https://makecode.microbit.org/offline-app>