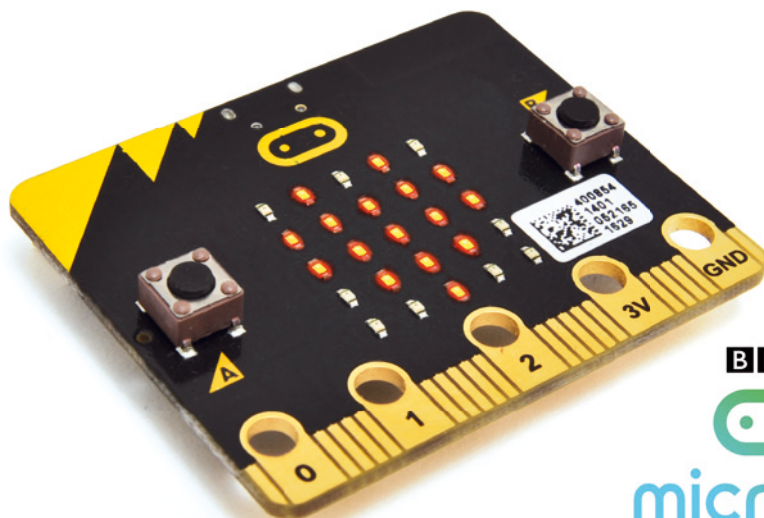


# Micro:bit



Micro:bit racchiude in una basetta un microcomputer a 32 bit, sensori di movimento, una bussola elettronica, un modulo Bluetooth, un buon numero di ingressi di I/O, alcuni pulsanti e numerosi LED disposti a matrice per lo sviluppo di interfacce grafiche. Per semplificare il compito a chi vuole muovere i primi passi nel campo della programmazione, è stato messo a disposizione un simulatore web che permette di scrivere e provare i propri programmi prima di scaricarli sul micro:bit. È possibile salvare sul proprio PC i programmi realizzati. Essendo il micro:bit essenzialmente un microcontrollore, stile Arduino, è possibile caricare ed eseguire un solo programma alla volta. Sono disponibili anche numerosi programmi di esempio e diversi tutorial, al momento unicamente in inglese.

## Caratteristiche

Il micro:bit dispone delle seguenti funzionalità ed interfacce, evidenziate nella **Fig. 1**

- 25 LED rossi che possono essere accesi con diverse intensità per creare disegni, animazioni e giochi;
- Due pulsanti che possono essere gestiti via programma per personalizzare il comportamento dello stesso;
- Un pulsante di reset e restart;
- Un accelerometro in grado di rilevare e riconoscere diversi tipi di movimento come lo scuotimento, le rotazioni e la caduta libera;
- Una bussola elettronica in grado di rilevare l'orientamento e la rotazione in gradi rispetto al Nord. Un uso ulteriore potrebbe essere l'utilizzo come rilevatore metalli (magnetici);
- Un connettore USB per collegare il

micro:bit al PC per caricare i programmi software;

- Un connettore per l'alimentazione esterna;
- Un modulo Bluetooth Smart Technology che permette di connettere il micro:bit con gli altri dispositivi dotati della stessa tecnologia come altri micro:bit, PC, smartphone, kit Bluetooth, tablet e fotocamere;
- Un sensore di temperatura in grado di rilevare la temperatura ambiente;
- Un connettore con 24 pin dei quali cinque con piazzole adatte ad essere collegate con morsetti a coccodrillo. Sul connettore sono disponibili I/O analogici e digitali, bus di comunicazione ed altre funzionalità descritte più dettagliatamente in seguito.

### Descrizione hardware

Integrate nel chip vi è una congrua serie di periferiche. Oltre a queste, nel micro:bit

sono presenti diverse periferiche aggiuntive. Per il trasferimento dei programmi da PC al micro:bit è presente un'interfaccia USB realizzata per mezzo del chip Freescale KL26Z.

### nRF51 Application Processor

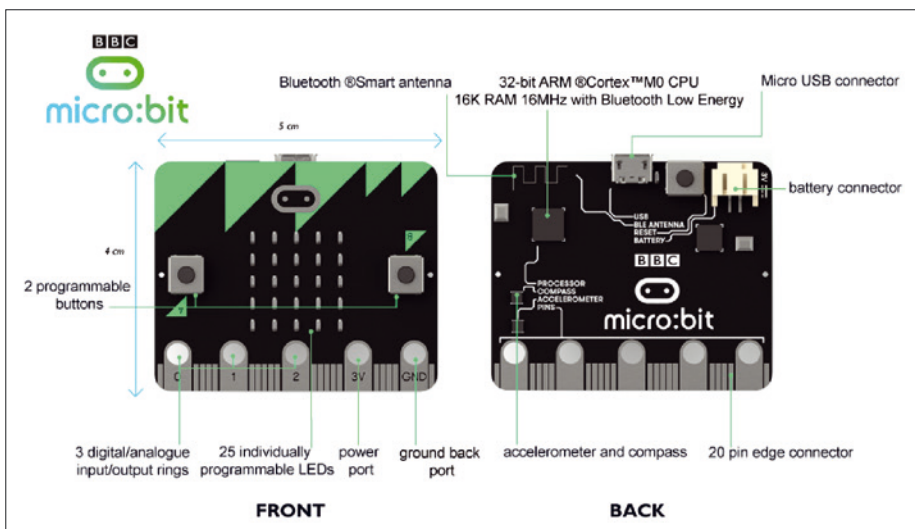
I programmi trasferiti da PC a micro:bit vengono eseguiti sul chip nRF51, sul quale è installato, in modo predefinito, il bootloader che opera il trasferimento dei programmi da PC e la loro esecuzione. Il programma trasferito ingloba in un unico file il codice scritto dall'utente, il codice run-time e lo stack Bluetooth.

Allo stesso SoC fanno capo i pin GPIO disponibili all'utente. In particolare il SoC include una CPU ARM Cortex M0 a 32 bit e 16 MHz di clock, 256KB di memoria ROM Flash e 16KB di memoria RAM.

### Comunicazione Bluetooth

La comunicazione Bluetooth del micro:bit

**Fig. 1**



è assicurata, dal punto di vista hardware, dal transceiver a 2,4 GHz e dal device software Nordic S100 SoftDevice, che fornisce uno stack completo per la comunicazione Bluetooth low Energy, in modo da permettere al micro:bit di comunicare con la più ampia gamma di dispositivi, inclusi smartphone e tablet. La comunicazione Bluetooth utilizza le bande di frequenza 2.4GHz ISM (Industrial, Scientific and Medical) 2.4GHz..2.41GHz su 40 (0 – 39) dei 52 canali a 2MHz disponibili. La minimizzazione della congestione sui canali avviene con la tecnica del Frequency Hopping.

### **Modulo radiofrequenza**

Tutta la comunicazione a radiofrequenza è gestita dal transceiver hardware a 2,4 GHz. Il protocollo Bluetooth è implementato via software. Il transceiver include in modo predefinito il protocollo proprietario Nordic Gazell, che permette lo scambio di informazioni con altri dispositivi che supportano lo stesso protocollo mediante piccoli pacchetti di dati molto semplici. La maggior parte dei linguaggi che possono essere utilizzati per programmare il micro:bit dispongono dell'interfaccia "radio" che è implementata utilizzando proprio il protocollo Nordic-Gazell. Un "group code" aggiunto automaticamente dal run time di micro:bit ai pacchetti di dati, permette di gestire una semplice forma di indirizzamento e filtraggio dei pacchetti. La banda di comunicazione è suddivisa in 101 canali (0 – 100). I messaggi vengono trasmessi in chiaro e possono essere di 32 byte o 255 byte di lunghezza. La potenza di trasmissione può essere configurata su otto livelli di potenza da -30dbm

(livello 0) a +4dbm (livello 7).

### **Pulsanti**

Nella parte frontale del micro:bit sono presenti due pulsanti. Un pulsante singolo è presente nella parte inferiore. I pulsanti sulla parte frontale sono riconosciuti come A e B, come confermato dalla serigrafia sul circuito stampato e possono essere utilizzati all'interno dei programmi per qualsiasi operazione. Il debouncing dei pulsanti è gestito dal run-time software e le condizioni riconoscibili sono la pressione breve, la pressione lunga e la pressione contemporanea di entrambi i pulsanti A e B. I pulsanti operano in logica inversa, ovvero quando il pulsante è in stato di rilascio una resistenza di pull-up mantiene l'ingresso corrispondente al livello logico "1", mentre quando il pulsante viene premuto, il contestuale collegamento a massa porta l'ingresso corrispondente al livello logico "0". I pulsanti A e B sono collegati rispettivamente ai pin P5 e P11 del GPIO, che sono disponibili anche sul connettore a pettine presente sul circuito stampato. Il pulsante sul retro del circuito stampato funziona principalmente da pulsante di reset ed è collegato sia al processore di interfaccia USB KL26 che al processore principale nRF51. La pressione del pulsante di reset ha l'effetto di reinizializzare l'applicazione, sia nel caso il micro:bit sia alimentato via USB che mediante la batteria esterna.

### **Display**

Il display del micro:bit è costituito da una matrice di LED rossi disposti su cinque file e cinque colonne. Il software run-time del micro:bit aggiorna lo stato di ciascun

LED a frequenza tale da non rendere lo sfarfallio percettibile all'occhio. Per ciascun LED appartenente alla matrice è possibile impostare la luminosità su una scala di dieci livelli. Una curiosa funzionalità è l'utilizzo della matrice di LED per misurare l'intensità della luce ambientale. Questo avviene da parte del software run-time commutando periodicamente alcuni pin della matrice di LED e misurando il tempo di caduta della tensione, che è grossolanamente proporzionale al livello di luce ambientale. Anche la misura della luminosità viene restituita su una scala di dieci valori.

### Sensori

Sul circuito stampato del micro:bit sono montati due circuiti integrati con funzione di sensori, un accelerometro ed un magnetometro. L'accelerometro misura le accelerazioni sui tre assi mentre il magnetometro può essere utilizzato sia come bussola che come rilevatore di campi magnetici. Entrambi i sensori sono collegati al controllore principale per mezzo del bus I2C che fa capo ai pin P19 (SCL) e P20 (SDA) del connettore GPIO. A questi pin sono collegate due resistenze di pull-up. Il magnetometro può generare un interrupt verso il processore principale mentre l'accelerometro può generare due differenti interrupt verso il processore principale.

### Accelerometro

L'accelerometro è in grado di misurare l'accelerazione sui tre assi. Il sensore è in grado di interpretare i dati provenienti dall'accelerometro per identificare alcuni comportamenti specifici direttamente in

hardware, come la caduta libera, o via calcoli software come l'orientamento del sensore (su o giù) e lo scuotimento. L'intergrato Freescale MMA8653FC, che realizza l'accelerometro colloquia con il processore principale per mezzo del bus I2C. Sono configurabili tre range di misurazioni 2, 4 o 8 g. La risoluzione della lettura è di dieci bit (0..1023). La massima frequenza di campionamento è di 800Hz.

### Magnetometro

Il magnetometro è realizzato con l'intergrato Freescale MAG3110, in grado di rilevare l'intensità di un campo magnetico. La trasformazione in dati di orientamento della scheda, a partire dai valori rilevati, viene eseguita da opportuni algoritmi inclusi nel software run-time. La bussola magnetica deve essere calibrata prima di essere utilizzata. Il processo di calibrazione è gestito dal software run-time. Il magnetometro comunica con il processore principale utilizzando il bus I2C. La massima frequenza di campionamento è di 80Hz. La massima intensità di campo magnetico misurabile è di 1000 uT (micro Tesla) con una sensibilità di 0.10 uT.

### Sensore di temperatura

Il processore principale nRF51 include un sensore di temperatura, che può essere interrogato attraverso il software run-time, e fornisce un valore approssimato della temperatura ambiente. L'intervallo di temperatura misurabile va da -25 °C a 75 °C, con una risoluzione di 0,25 °C ed una accuratezza di +/- 4 °C.

### Connettore a pettine

Il connettore a pettine è l'interfaccia prin-

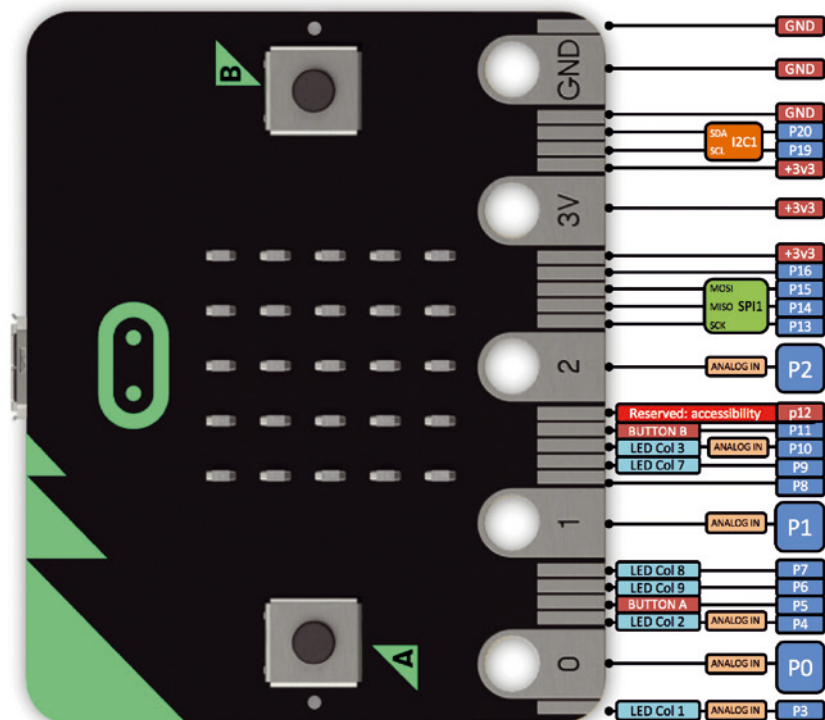


Fig. 2

principale del micro:bit verso il mondo fisico e fornisce una varietà di I/O digitali, analogici, touch, pwm, oltre ai bus di comunicazione seriale, I2C e SPI. I terminali del connettore a pettine hanno un passo di 1,27 mm tranne alcune piazzole, che sono più larghe delle altre e possiedono un foro centrale di 4 mm al quale può essere collegato un terminale a coccodrillo, come visibile in Fig. 2. In particolare ai terminali P0, P1 e P2 sono collegate delle resistenze di pull up da 10 Mohm, che permettono di abilitare i terminali come sensori touch. In questa configurazione i terminali sono normalmente a livello alto. Quando si tocca contemporaneamente un terminale e

la massa, il terminale interessato viene portato a livello basso (0V). Nell'utilizzo normale, quindi in modalità non touch, su questi terminali viene abilitata via software una resistenza di pull down in modo da tenere i terminali a livello basso quando si trovano in stato di riposo e portarli a livello alto quando collegati al terminale positivo. Tutti i terminali da un lato del circuito stampato sono elettricamente connessi ai corrispondenti terminali sul lato opposto. Per evitare corti circuiti o generare interferenze, i terminali a destra e a sinistra delle piazzole grandi GND e 3V sono anch'essi collegati rispettivamente a massa (GND) e al positivo della tensione (3V). Ai terminali

sul connettore possono essere assegnate funzioni differenti impostabili via software, a seconda delle esigenze dell'utilizzatore. Sul connettore a pettine sono riportate molte delle funzionalità di comunicazione e di I/O messe a disposizione dal processore principale nRF51. Molte delle funzioni assegnate in modo predefinito possono essere disabilitate via software per trasformare i rispettivi terminali in ingressi/uscite di utilizzo generale. Una nota importante riguarda la quantità di corrente che può essere prelevata dal connettore a pettine. I pin di I/O possono essere configurati sia in modalità std-drive che in modalità high-drive. In modalità std-drive ciascun pin può erogare fino a 0,5mA di corrente mentre in modalità high-drive possono essere erogati fino a 5mA con il vincolo di limitare a tre il numero di terminali configurati contemporaneamente in high-drive mode. Infine la risoluzione della conversione ADC è di 10 bit (0..1023).

### **Alimentazione del micro:bit**

È possibile alimentare il micro:bit da tre sorgenti differenti: tramite il connettore USB, con una batteria esterna collegata all'apposito connettore e collegando una sorgente di alimentazione ai terminali 3V e GND presenti sul connettore a pettine. Nel caso di alimentazione tramite USB la tensione viene regolata ai valori corretti per il micro:bit dal regolatore interno al processore di interfaccia KL26. Il circuito di commutazione tra le sorgenti di alimentazione utilizza un diodo low-Vf (max 0.23V) per prevenire i ritorni di tensione tra le diverse sorgenti. Bisogna prestare particolare attenzione se si decide di alimentare il micro:bit utilizzando i termina-

li 3V e GND del connettore a pettine. Il terminale 3V è collegato direttamente agli integrati presenti sulla scheda ed una tensione superiore ai 3,6V può danneggiare irrimediabilmente il micro:bit. Il terminale 3V può essere utilizzato per alimentare piccoli circuiti esterni (corrente max 90mA).

### **Comunicazione USB**

La comunicazione con il micro:bit via USB (versione 1.1 a 12Mbit) è gestita via software con un apposito stack codificato nel firmware dell'integrato di interfaccia KL26, che include una CPU ARM Cortex M0 a 32 bit e 16MHz di clock, 32KB di ROM Flash e 4KB di RAM. Lo stack software permette di vedere il micro:chip da un normale PC come una memoria di massa esterna, tipo chiavetta USB. Questa funzionalità permette di trasferire il software dal PC al micro:bit semplicemente trascinando il file dalla cartella di origine alla cartella radice del micro:bit. È anche possibile utilizzare la connessione USB come connessione seriale per la trasmissione e ricezione dati da dispositivi esterni come altri micro:bit, microcontrollori e PC.

### **BBC micro:bit runtime**

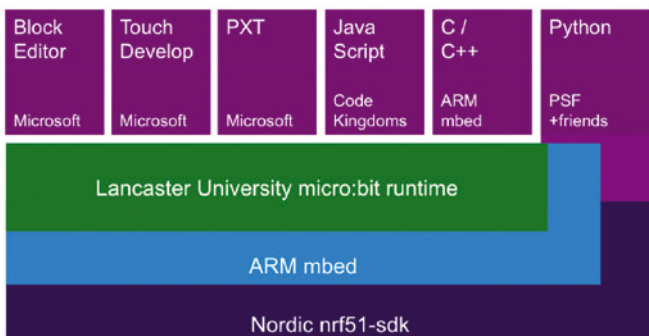
La flessibilità e semplicità di programmazione del micro:bit è garantita dalla presenza del software run-time, una architettura software in grado di "interpretare" ed eseguire correttamente i programmi scritti nei vari linguaggi supportati. Il runtime è composto da diversi strati software che permettono, nella loro globalità, di semplificare la programmazione del micro:bit utilizzando i diversi linguaggi di programmazione ed ambienti di sviluppo

disponibili per il micro:bit come Microsoft Block Editor, Code Kingdom's JavaScript, TouchDevelop e PXT. Altri linguaggi di sviluppo e ambienti di programmazione preesistevano rispetto alla realizzazione del micro:bit e sono stati adattati all'utilizzo con quest'ultimo come ARM mbed e MicroPython. L'architettura del run-time è visibile in **Fig. 3**. Lo strato più basso è il livello di astrazione dall'hardware (HAL – Hardware Abstraction Layer) fornito dall'ambiente ARM mbed. ARM mbed fornisce uno strato software utile agli sviluppatori per scrivere software adatto ad essere eseguito su tutta la famiglia dei processori ARM Cortex, come il Nordic nRF51822, utilizzato nel micro:bit e standardizza l'accesso e l'utilizzo alle periferiche e risorse hardware come il GPIO ed i bus seriale, SPI ed I2C. Su questo strato la Lancaster University ha costruito il run-time vero e proprio, che fornisce una serie di interfacce di alto livello ad uso degli utilizzatori (gli studenti delle scuole), per interfacciare in modo semplice e standard le funzionalità del micro:bit con i diversi linguaggi di sviluppo previsti. La maggior parte dei “blocchi” utilizzati nella programmazione “didattica”, come quelli

presenti in Microsoft Blocks o PXT, richiamano funzioni messe a disposizione del run-time.

I componenti principali del run-time del micro:bit sono:

- Uno schedatore che permette al micro:bit di eseguire diversi task contemporaneamente (o quasi);
- Un sistema di gestione degli eventi denominato “message bus” che permette di scrivere codice basato su eventi. Questo sistema permette di comunicare al codice scritto dall'utilizzatore cosa sta accadendo sulle interfacce con il mondo fisico esterno; dalla pressione di un pulsante ad un messaggio ricevuto via radio;
- Una serie di device driver che permettono di comunicare con la maggior parte delle periferiche hardware presenti sul micro:bit compresi il display a LED, i diversi sensori, il file system, i profili radio e bluetooth;
- Una serie di tipi dati che permettono di rendere trasparente la gestione della memoria per la memorizzazione dei dati applicativi. Questa funzionalità, implementata originariamente per essere utilizzata con i linguaggi di alto livello è molto utile anche per chi vuole cimentarsi nella pro-



**Fig. 3**

The screenshot shows the micro:bit website homepage. At the top, there is a navigation bar with the micro:bit logo and links for 'About', 'Get Started', 'Let's Code', 'Ideas', 'Teach', 'Education', and a search icon. A 'Language' dropdown menu is also visible. The main heading reads 'Power your imagination with code'. Below this, several programming environments are featured:

- Microsoft PXT (Beta)**: Described as a new programming experience supporting both a block-based editor and JavaScript. A button says 'Try the PXT beta'.
- Python**: A text-based editor with auto-complete and pre-made images/music. Buttons include 'Start with this editor', 'Documentation', and 'Lessons'. A button below says 'Try the beta editor with radio'.
- Microsoft Block Editor**: A visual editor using drag-and-drop blocks. Buttons include 'Start with this editor', 'Documentation', and 'Lessons'.
- Javascript**: A visual editor with a drag-and-drop interface. A button says 'Start with this editor'.
- Microsoft Touch Develop**: A touch-based interface for mobile devices. A button says 'Start with this editor'.

At the bottom, there is a 'Mobile Apps' section with 'GET IT ON Google Play' and 'Download on the App Store' buttons. A paragraph explains that the mobile app lets users send code to their micro:bit via Bluetooth. The footer contains links for 'Terms of use', 'Privacy policy', 'Our Team', 'Contact', 'Support', and 'Developer Community', along with the copyright notice '© Copyright Micro:bit Educational Foundation'.

Fig. 4



grammazione del micro:bit in C e C++.

Una nota per chi vuole provare a programmare il micro:bit in C o C++. Questa operazione è possibile in due modi. È possibile utilizzare direttamente l'HAL mbed (Hardware Abstraction Layer) per poter accedere alle periferiche a basso livello oppure, come caldamente raccomandato, utilizzare il run-time che espone una serie di API che possono essere utilizzate con i programmi C e C++. L'interfaccia API è stata realizzata in modo modulare e quindi possono essere inclusi nei programmi C e C++ solo i moduli effettivamente necessari, per esempio solo il modulo necessario a gestire il display LCD.

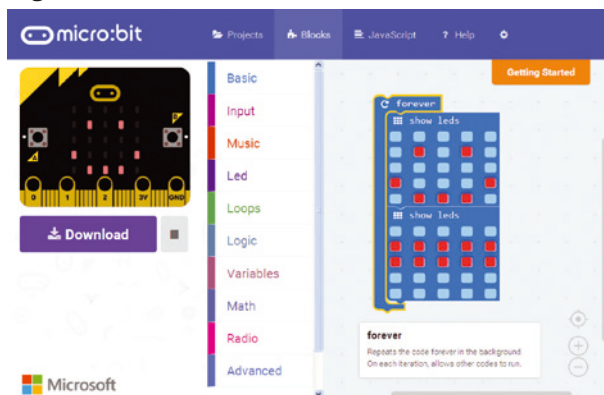
È possibile trovare abbondante documentazione sul run-time del micro:bit all'indirizzo:

<https://lancaster-university.github.io/microbit-docs/>

### Programmazione

Dopo aver caricato nella cache del browser le applet degli ambienti di sviluppo, è possibile scrivere i propri programmi anche in assenza di un collegamento internet. Per i dispo-

**Fig. 5**



sitivi mobili, come tablet e smartphone, oltre che gli ambienti web sopra citati sono disponibili anche "app" gratuite scaricabili dagli store dei provider.

Prima di iniziare dobbiamo collegare il micro:bit al nostro PC utilizzando un cavo USB. Una volta collegato, micro:bit dovrebbe venire riconosciuto immediatamente. Nel caso di

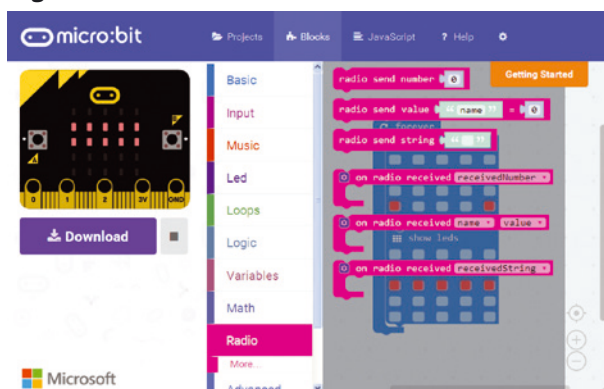
Windows XP verrà richiesto di installare il driver.

Ora collegatevi al sito di riferimento di micro:bit all'indirizzo:

<http://microbit.org/>

Nella pagina principale cliccate sul pulsante "Let's Code" e vi troverete nella pagina visibile in **Fig. 4** che elenca i diversi am-

**Fig. 6**

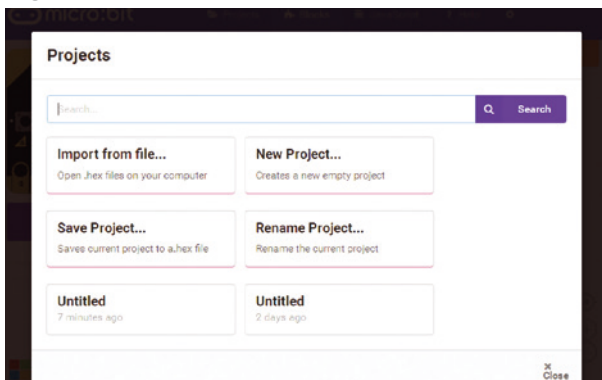


bienti di programmazione. Vediamoli in dettaglio.

### Microsoft PXT (Beta)

Microsoft PXT (Programming Experience Toolkit) è un ambiente di programmazione realizzato da Microsoft (ancora in versione beta) che supporta sia un ambiente grafico a blocchi, in stile Scratch, che il linguaggio Javascript. Sulla sinistra dell'editor è presente la collezione dei blocchi suddivisi per categorie. Per comporre un programma si deve scegliere il blocco desiderato dalla collezione che lo contiene, e trascinandolo nello spazio centrale. Degno di nota è il blocco che rappresenta l'intero display. Cliccando sui singoli "LED" si cambia la condizione da acceso a spento e viceversa come visibile nell'esempio in **Fig. 5**. Da notare il simulatore in alto a sinistra che può essere attivato dal pulsante con l'icona a freccia o a quadrato, a seconda dello stato. Dopo aver composto un programma basta cliccare su uno spazio bianco nel pannello dell'editor per vedere il programma in esecuzione sul simulatore. Altre funzionalità degne di nota

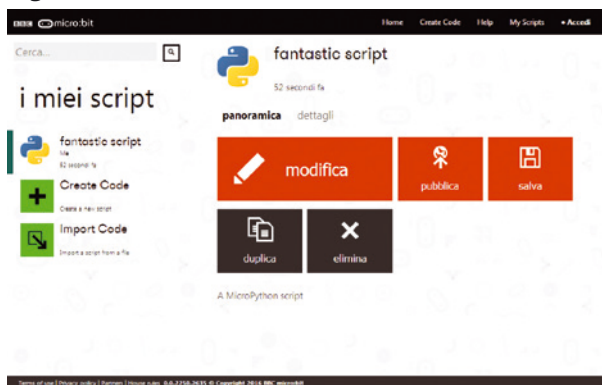
**Fig. 7**



**Fig. 8**



**Fig. 9**



sono quelle che permettono la comunicazione in radiofrequenza tra due o più

micro:bit (**Fig. 6**). Per tutti i blocchi è possibile aprire una finestra di aiuto clic-

candovi sopra col tasto destro del mouse. Per creare il file da trasferire sul micro:bit cliccate sul pulsante “Download”. Vi verrà chiesto di salvare il file. Scegliete una cartella ed eseguite il salvataggio. Il metodo per trasferire il programma nel micro:bit è identico per tutti gli ambienti di sviluppo. Cliccando sul pulsante “Projects”, in alto a sinistra si apre la finestra visibile in **Fig. 7** che permette di salvare i “sorgenti” in sviluppo sul proprio PC e di importare da PC ad ambiente di sviluppo.

### Python

Python conosciuta come MicroPython. MicroPython è stato ed è tuttora in sviluppo per permettere la programmazione dei microcontrollori in un linguaggio compatibile con Python. Il metodo consiste nello scrivere il programma con un editor e poi “iniettarlo” nel microcontrollore dove una macchina virtuale dedicata provvede ad interpretarla ed eseguirlo. Nel caso del micro:bit queste operazioni sono a carico del run-time. L'opportunità di utilizzare il linguaggio Python permette di dare continuità, in un ipotetico percorso didattico, verso l'utilizzo di sistemi più complessi come i microcomputer in grado di ospitare GNU/Linux, come la serie Raspberry Pi. Per aprire l'ambiente Python, sempre dalla pagina visibile in **Fig. 4** cliccate sul pulsante “Start with this editor”, nella sezione “Python”. Si apre l'editor di programmazione Python. Per fare una prova modificate la scritta come visibile in **Fig. 8**.

Salvate il programma in un file cliccando sull'icona “Download”.

Una documentazione completa sulle istruzioni disponibili nel linguaggio Python per il micro:bit è raggiungibile all'indirizzo:

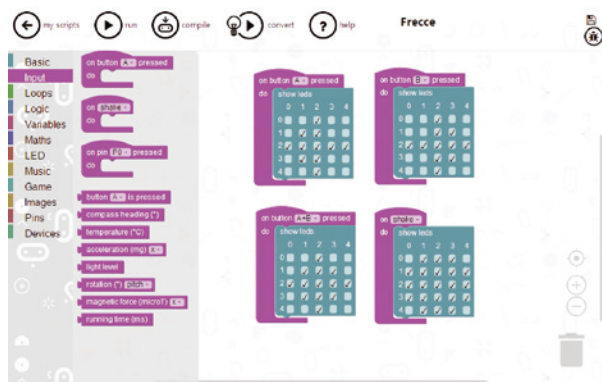
<https://microbit-micropython.readthedocs.io/en/latest/>

Cliccando sul pulsante “Snippets” si accede ad una nutrita libreria di esempi e di frammenti di codice riutilizzabile nei propri progetti. Cliccando sul simbolo “freccia” in alto a sinistra si apre la finestra visibile in **Fig. 9** che permette di salvare i “sorgenti” in sviluppo sul proprio PC e di importare da PC ad ambiente di sviluppo i programmi salvati in precedenza. È inoltre possibile passare da un ambiente di sviluppo all'altro. Infatti questa interfaccia è condivisa dagli ambienti di sviluppo che descriviamo in seguito.

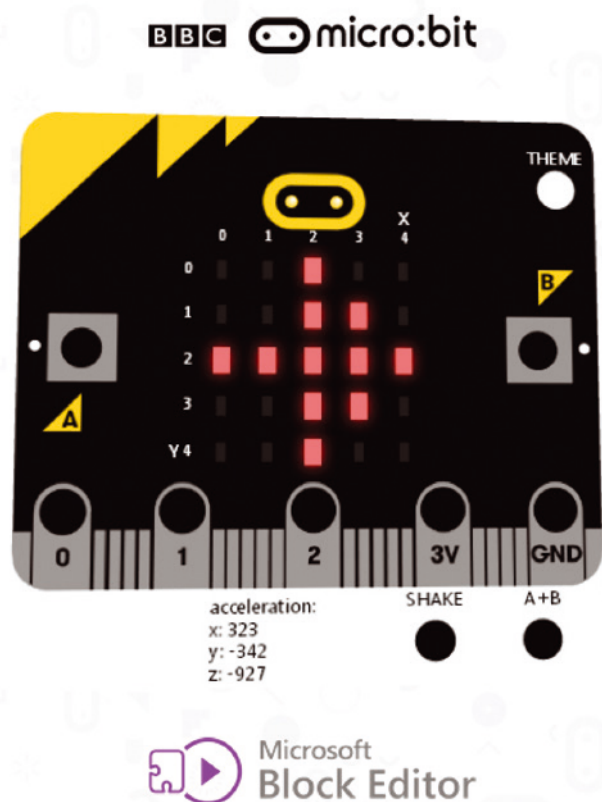
### Microsoft Block Editor

Il Microsoft Block Editor è un editor visuale e rappresenta il punto di partenza per chi vuole iniziare a programmare. Permette di approfondire la programmazione strutturata e la gestione di eventi. Il metodo di composizione di un programma consiste nello scegliere i blocchi che rappresentano le istruzioni, trascinarli nel pannello dell'editor, configurarli nelle parti personalizzabili ed “incastrarli” in modo corretto. In questo si è facilitati dalla forma dei blocchi. I blocchi si connettono solo se sono compatibili e le forme corrispondono. Chiaramente questo non garantisce che la logica che si va a realizzare sia quella desiderata. Per questo dobbiamo fare una buona analisi di ciò che si desidera realizzare e tradurlo in algoritmi in modo coerente. Anche in questo ambiente di sviluppo vi sono blocchi molto “espressivi”, come quello disponibile per configurare il display a LED, esattamente come nel caso dell'ambiente Microsoft PXT. Questa similitudine tra le interfacce

grafiche e testuali degli ambienti di programmazione disponibili permette un passaggio graduale e incrementale dai linguaggi con interfaccia grafica ai linguaggi tradizionali con interfaccia testuale. Anche in questo ambiente di sviluppo è possibile eseguire il codice appena scritto in ambiente di simulazione, per eseguire i test di correttezza, prima di trasferire il codice stesso sul micro:bit fisico. Per attivare l'ambiente di simulazione si deve cliccare sull'icona "Play", la seconda in alto da sinistra. Una volta entrati nell'ambiente di simulazione possiamo provare le diverse funzionalità. Nell'esempio visibile in **Fig. 10** abbiamo provato a realizzare un piccolo sistema ad eventi che intercetta la pressione dei pulsanti A e B, singoli o premuti contemporaneamente, oltre allo scuotimento. Ciascuna condizione viene evidenziata con una diversa configurazione della matrice di LED. Nell'ambiente di emulazione visibile in **Fig. 11** viene presentata l'immagine del micro:bit con i pulsanti "funzionanti", un pulsante separato per la pressione contemporanea



**Fig. 10**



**Fig. 11**

di A e B ed un pulsante per simulare lo scuotimento. Muovendo il micro:bit con il mouse vengono evidenziati i valori di accelerazione sui tre assi. Nell'ambiente di sviluppo Block Editor sono presenti diversi "blocchi" che permettono di rilevare lo stato ed i valori dei sensori. L'icona con il simbolo del micro:bit permette di scaricare il codice sul proprio PC per poi trasferirlo al micro:bit. Il pulsante con l'icona a "punto interrogativo" apre l'help con la descrizione di tutti i blocchi e le rispettive modalità di utilizzo.

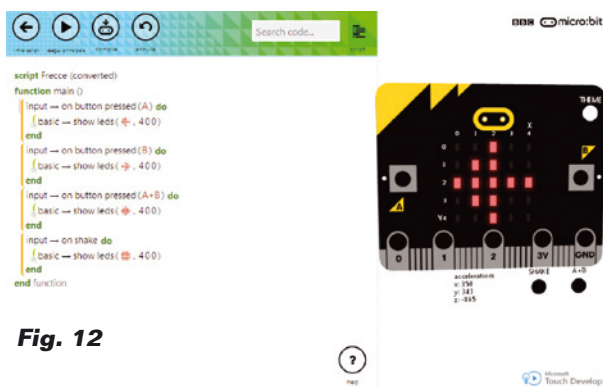


Fig. 12

**Microsoft Touch Develop**  
L'ambiente di sviluppo Microsoft Touch Develop è stato sviluppato per realizzare i programmi per il micro:bit utilizzando tablet od altri dispositivi dotati di schermo touch. Ovviamente può essere utilizzato anche dai PC normali utilizzando tastiera e mouse. Il linguaggio di Touch Develop è un linguaggio di scripting con una sintassi fortemente strutturata dove l'editor guida la stesura dei blocchi di istruzioni e le singole linee di codice. Cliccando nell'area delle istruzioni si

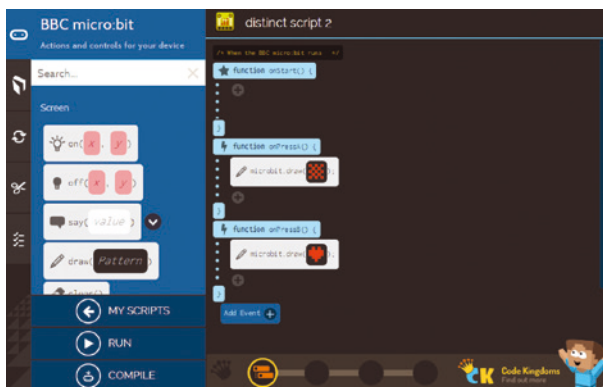


Fig. 13

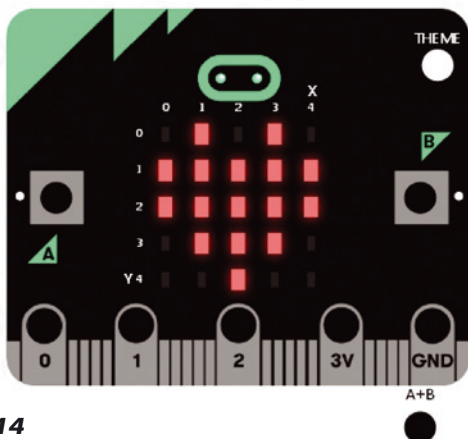


Fig. 14

apre in basso un box contestuale che mostra le diverse opzioni possibili nel punto del codice che si sta editando. Toccando o cliccando in sequenza sui diversi blocchetti di compone ciascuna riga di codice ed i blocchi di istruzioni. L'esempio che vedete in **Fig. 12** è stato generato convertendo l'esempio scritto con il Microsoft Block Editor. Per eseguire la conversione cliccate sul pulsante con la lampadina ed il triangolino rivolto verso destra (Fig. 12). Nella pagina che si apre cliccate sul pulsante "Convert".

Vi troverete nell'ambiente di Touch Develop con il codice convertito sulla sinistra. Anche in questo caso è disponibile l'ambiente di simulazione, che si attiva cliccando sul pulsante a forma di triangolo con la punta rivolta a destra (secondo pulsante in alto a sinistra). Per scaricare il programma da eseguire sul micro:bit si clicca sul pulsante con il simbolo del micro:bit. A fianco trovate il pulsante "Undo". Cliccando sul primo pulsante in alto a sinistra, quello con la freccia rivolta a sinistra, si entra nella solita pagina dove è possibile salvare il sorgente del programma su PC e importare nell'ambiente di sviluppo un programma salvato precedentemente.

### Javascript

L'ambiente di sviluppo Javascript permette, a chi ha fatto progressi nella programmazione, di sviluppare programmi utilizzando la sintassi testuale di Javascript, con un editor guidato che supporta il metodo "clicca e trascina", per la composizione delle righe e dei blocchi di codice, come abbiamo provato a fare nell'esempio di **Fig. 13**. Cliccando sul pulsante

"RUN" si attiva l'ambiente di simulazione visibile in **Fig. 14**. Per salvare su PC il programma da trasferire sul micro:bit cliccate sul pulsante "COMPILE". Per salvare il sorgente o importare programmi salvati in precedenza andate alla pagina "i miei script" cliccando sul pulsante "MY SCRIPTS".

### Eseguire i programmi sul micro:bit

Abbiamo visto le caratteristiche tecniche del micro:bit e i diversi ambienti di sviluppo che permettono di programmarlo e di verificare il funzionamento per mezzo dei simulatori messi a disposizione. Vediamo ora come trasferire ed eseguire i nostri programmi sul micro:bit reale. Per prima cosa dobbiamo procurarci un micro:bit e collegarlo al nostro PC tramite il cavetto USB. Quando viene collegato per la prima volta, di solito, il micro:bit viene riconosciuto automaticamente. In caso ciò non avvenga e viene aperta la finestra di configurazione del driver, basta scegliere l'opzione "Ricerca un driver su internet". Una volta collegato e riconosciuto il micro:bit viene "visto" come una memoria di massa esterna, in pratica come una chiavetta USB. A questo punto, per caricare un "programma" sul micro:bit basta portarlo in una cartella del nostro PC e da qui trasferirlo nel file system del micro:bit. Nella descrizione dei singoli ambienti di sviluppo abbiamo indicato come ottenere il file "eseguibile" del nostro programma e come salvarlo in una cartella del nostro PC. A questo punto copiate il file dalla cartella dove lo avete salvato e incollatelo nel file system del micro:bit, oppure trascinatelo semplicemente dalla cartella di origine al micro:bit. Durante il trasferimen-

to del file vedrete lampeggiare il LED sulla faccia posteriore del micro:bit.

Una volta terminato il trasferimento il programma entrerà in funzione immediatamente, come avviene normalmente con i microcontrollori.

A questo punto il programma rimane memorizzato in permanenza nella memoria flash. Se spegnete (togliendo alimentazione) e riaccendete il micro:bit, o premete il pulsante reset, verrà eseguito l'ultimo programma caricato. Questo anche se il micro:bit viene alimentato a batteria o tramite i terminali 3V e GND del connettore a pettine. Il micro:bit può essere programmato con un solo programma alla volta. Può capitare di tentare di caricare un programma .hex nel file system del micro:bit quando ne è già presente un altro. In questo caso il trasferimento non va a buon fine. Per rimediare basta togliere e ridare alimentazione al micro:bit. Il file presen-

te in precedenza sparirà dal file system e sarà possibile caricarne uno nuovo. In definitiva abbiamo tra le mani un ottimo strumento per imparare a programmare con diversi ambienti di sviluppo con crescenti livelli di difficoltà che ci permetteranno di acquisire ed approfondire le più disparate tecniche di programmazione e di utilizzo di periferiche, sensori, bus di comunicazione via cavo o a radiofrequenza.

L'articolo completo del progetto è pubblicato su Eletttronica In n. 213

**A tutti i residenti nell'Unione Europea****Importanti informazioni ambientali relative a questo prodotto**

Questo simbolo riportato sul prodotto o sull'imballaggio, indica che è vietato smaltire il prodotto nell'ambiente al termine del suo ciclo vitale in quanto può essere nocivo per l'ambiente stesso. Non smaltire il prodotto (o le pile, se utilizzate) come rifiuto urbano indifferenziato; dovrebbe essere smaltito da un'impresa specializzata nel riciclaggio. Per informazioni più dettagliate circa il riciclaggio di questo prodotto, contattare l'ufficio comunale, il servizio locale di smaltimento rifiuti oppure il negozio presso il quale è stato effettuato l'acquisto.

Distribuito da:

**FUTURA GROUP SRL**

Via Adige, 11 - 21013 Gallarate (VA) Tel. 0331-799775 Fax. 0331-792287

web site: [www.futurashop.it](http://www.futurashop.it)

supporto tecnico: [www.futurashop.it/Assistenza-Tecnica](http://www.futurashop.it/Assistenza-Tecnica)