

1 Passaggi Fondamentali

Questa sezione fungerà da introduzione all'editor Make:Code e ad un argomento fondamentale nella programmazione di microcontrollori: *Input/Output*.

1.1 Make:Code

Make:Code è un editor online per programmare MicroBit. Per accedervi basta andare sul sito <https://makecode.microbit.org/> e cliccare su *Nuovo Progetto*.

Il linguaggio a blocchi è simile a scratch, ma non uguale. Scratch nasce con l'obiettivo di insegnare a programmare tramite teatralizzazione e storytelling mentre Make:Code nasce con l'obiettivo di insegnare a programmare usando un microcontrollore e gamification.

Questa differenza è evidente nel modo in cui l'editor invita a strutturare il codice, scratch invita a creare molti codici "in parallelo" utilizzando un punto di partenza per ogni sequenza di comandi, makecode, invece, invita una struttura del codice sequenziale.

La struttura del codice in makecode è molto simile a quella di un programma per microcontrollori scritto in un linguaggio di programmazione testuale, ovvero composta da due blocchi di codice, il "setup" e il "loop".

Il primo blocco racchiude il codice che viene eseguito una sola volta all'avvio del microcontrollore, mentre il secondo blocco racchiude il codice che viene eseguito in loop, ovvero ripetutamente, fino a quando il microcontrollore è acceso.

È possibile programmare Micro:Bit utilizzando scratchX e l'estensione per Micro:Bit, ma non è consigliato in quanto non è possibile utilizzare tutte le funzionalità di Make:Code e non si ha a disposizione il simulatore.

1.1.1 [Attività] Discussione sulle differenze tra Scratch e MakeCode

Questo potrebbe essere un ottimo momento per raccogliere delle osservazioni informali osservando come il gruppo si approccia ad un nuovo editor, sia in fase di "gioco" che in quella di confronto.

L'obiettivo di questa attività è quello di far discutere gli studenti sulle differenze tra Scratch e MakeCode.

Per fare ciò si consiglia di dividere gli studenti in gruppi da 2-3 persone e dare 5 minuti di tempo per giocare con l'editor e capire come funziona.

In seguito dare 10 minuti agli studenti e chiedere loro di identificare i punti in comune e le differenze tra Scratch e MakeCode (volendo si può fornire un esempio prima di iniziare).

Scaduto il tempo chiedere alla classe di esporre le proprie idee e discuterle.

Esempi

- Scratch ha una finestra in cui si possono vedere ed animare i personaggi, MakeCode ha una finestra con un simulatore del microcontrollore.
- La categoria "Sensori" di scratch corrisponde alla categoria "Input" di MakeCode.
- Il comando "Dire" di scratch corrisponde al comando "Mostra Stringa/Numero" di MakeCode.

1.2 Input e Output

MicroBit presenta molte periferiche di input e output, come visto nelle sezioni precedenti.

In questa sezione verranno introdotti i concetti di input e output e verranno mostrati alcuni esempi di come utilizzarli. Le seguenti attività sono pensate per consentire agli studenti di familiarizzare con l'editor di makecode, con il simulatore e con i concetti di input e output relativi a microbit.

1.2.1 Attività 1 - Display

MicroBit ha un display a matrice di led che può essere utilizzato per mostrare testo, numeri e immagini.

L'obiettivo di questa attività è quello di scrivere un programma che scriva sul display il contenuto di una variabile.

Il valore verrà impostato nel blocco di setup e sarà costante per tutto il programma.

Soluzione Sono possibili più soluzioni, l'importante è che rispettino i seguenti criteri:

- Il valore della variabile deve essere impostato nel blocco di setup.
- Deve essere usato il comando corretto tra "Mostra Stringa" e "Mostra Numero" per mostrare il valore in base al tipo di dato salvato nella variabile.

Discussione *Il comando "Mostra Stringa/Numero" può essere inserito sia nel blocco di setup che in quello di loop, ci sono differenze? Cambia qualcosa se la stringa è composta da un solo carattere o da più caratteri? (Lo stesso discorso vale per il numero)*

Si consiglia di chiedere agli studenti cosa si aspettano che succeda e di far scrivere il codice per verificare le loro ipotesi.

Se il comando "Mostra Stringa/Numero" si trova nel blocco di setup, il display mostrerà solo una volta il contenuto della variabile. Se invece si trova nel blocco di loop, il display mostrerà il contenuto della variabile ripetutamente fino a quando il microcontrollore è acceso.

Il display di MicroBit può mostrare un carattere alla volta, per questo motivo il comando "Mostra Stringa/Numero" agirà in maniera diversa se la stringa è composta da un solo carattere o da più

caratteri.

Nel primo caso verrà visualizzato il carattere e questo resterà nel display fino a quando un nuovo comando non modificherà il display (In questo caso mettere il comando nel blocco di setup o loop è indifferente).

Nel secondo caso il display visualizzerà la sequenza di caratteri a scorrimento e, una volta terminata, il display tornerà vuoto. (In questo caso vi è una differenza tra mettere il comando nel blocco di setup o in quello di loop)

1.2.2 Attività 2 - Contatore

Esistono due modi di interagire con un pulsante, il primo è quello di controllare se il pulsante è premuto in un dato momento (se in questo momento il pulsante A è premuto, allora ..., altrimenti ...), il secondo consiste nell'utilizzare gli eventi (Quando A viene premuto esegui ...).

La principale differenza è che il primo metodo funziona in maniera sequenziale rispetto al resto del codice (se A viene premuto ma il codice non è arrivato al controllo allora la pressione viene ignorata), il secondo agisce in parallelo rispetto al resto del codice (Il codice viene eseguito immediatamente ogni volta che A viene premuto).

L'obiettivo di questa attività è quello di estendere il programma precedente al fine di aumentare o diminuire il valore di una variabile numerica mostrata a schermo in base al pulsante premuto.

Specifiche Il programma deve

- Mostrare il valore di una variabile numerica a schermo.
- Decrementare il valore della variabile di 1 se il pulsante A viene premuto.
- Incrementare il valore della variabile di 1 se il pulsante B viene premuto.
- Aggiornare costantemente il valore mostrato a schermo per riflettere quello della variabile.
- (OPZIONALE) Fare in modo che il valore non esca dal range $0 \leq x \leq 9$

Soluzione La soluzione dovrebbe estendere il codice scritto precedentemente (utilizzando "Mostra numero" nel blocco loop) aggiungendo due punti di partenza per la rilevazione degli eventi "Pulsante A premuto" e "Pulsante B premuto".

Si può trovare una possibile soluzione all'interno della cartella *sorgenti* con il nome *microbit-1-input_output.hex*¹

È possibile una seconda soluzione che prevede il comando "Monstra numero" a seguito di ogni incremento / decremento, consentendo così di non avere il blocco loop.

¹Per vedere come caricare un file in MakeCode consultare ??

1.3 Time

Nell'ambito della programmazione di microcontrollori è raro avere accesso al tempo reale (orario dell'orologio), generalmente si ha accesso ad un timer interno al microcontrollore che viene inizializzato all'avvio e può essere utilizzato per misurare intervalli di tempo.

1.3.1 Attività 3 - Cronometro

L'obiettivo di questa attività è quello di scrivere un programma che misuri il tempo trascorso tra due pressioni di un pulsante e lo mostri a schermo.

Specifiche Il programma deve:

- Mostrare un simbolo sul display per indicare che il cronometro è pronto.
- Alla prima pressione del pulsante A iniziare a misurare il tempo trascorso e cambiare il simbolo sullo schermo.
- Alla successiva pressione del pulsante A deve terminare la misurazione del tempo e scrivere il risultato sullo schermo

Soluzione Esistono diverse soluzioni a questo problema, nella cartella *sorgenti* sono presenti due delle soluzioni possibili, *microbit-2-time-a.hex* e *microbit-2-time-b.hex*.

La prima soluzione utilizza un comando "pausa" nel blocco di loop per rallentare il programma ed ogni secondo incrementa di 1 la variabile contatore.

La seconda soluzione salva in una variabile i millisecondi passati dall'avvio del programma alla prima pressione di A, poi salva quelli passati dall'avvio alla seconda pressione di A, infine calcola la differenza tra i due valori e la mostra a schermo.

La seconda soluzione è preferibile in quanto non rallenta il programma e non richiede di utilizzare un comando "pausa" ma è meno intuitiva dal punto di vista dello studente.

Discussione (Dopo aver mostrato entrambe le soluzioni) *Quali sono i vantaggi e gli svantaggi delle due soluzioni? Qual'è la soluzione più precisa?*

La prima soluzione è più semplice da capire e da implementare dato che usa meno variabili ma è meno precisa. Spesso si cerca di evitare il comando "pausa" nella programmazione di microcontrollori poichè questo interrompe l'esecuzione dell'intero programma (non è detto che sia sbagliato usarlo, dipende molto da cosa si sta creando).

Inoltre la prima soluzione è meno precisa in quanto l'incremento viene fatto ogni secondo, questo significa che se il tempo trascorso è 1.9 secondi verrà mostrato 1 secondo, se il tempo trascorso è 2.1 secondi verrà mostrato 2 secondi.

La seconda soluzione è più precisa in quanto utilizza i millisecondi per misurare il tempo trascorso, inoltre non blocca l'esecuzione del programma. Lo svantaggio principale di questa soluzione è l'introduzione di due variabili aggiuntive e la conseguente complicazione del codice.