



Misurare i dati ambientali: luce, rumore, temperatura

Materiali e strumenti necessari

Per programmare una scheda micro:bit in modo che raccolga umidità e temperatura, avrai bisogno del seguente hardware:

- Scheda Micro:bit V2 e relativi sensori integrati: la scheda principale programmabile comprende un sensore di luce integrato tramite il suo display LED, un sensore di livello sonoro integrato e un sensore di temperatura integrato - Circa 19 EUR per micro:bit
- Cavo micro-USB: per alimentare e programmare il micro:bit
- Batteria esterna (opzionale): per il funzionamento portatile se è necessario staccare il micro:bit. Troverai la custodia ufficiale per la batteria del micro:bit disponibile per l'acquisto a circa 2,20 EUR a confezione.

È anche possibile acquistare il kit Micro:bit V2 comprensivo di cavo USB e batteria al prezzo di 21 EUR per kit o 177 EUR per 10 kit.

- Computer o tablet: per scrivere e caricare il codice.
- Ambiente di programmazione: editor online MakeCode



Per questa fase, consigliamo di programmare almeno da 3 a 6 schede micro:bit da condividere tra gli studenti e raccogliere più informazioni e dati. È possibile farlo con una singola scheda, ma sarà necessario estendere il periodo di raccolta complessivo o ridurre il periodo di raccolta per studente da 7 a 3 giorni.

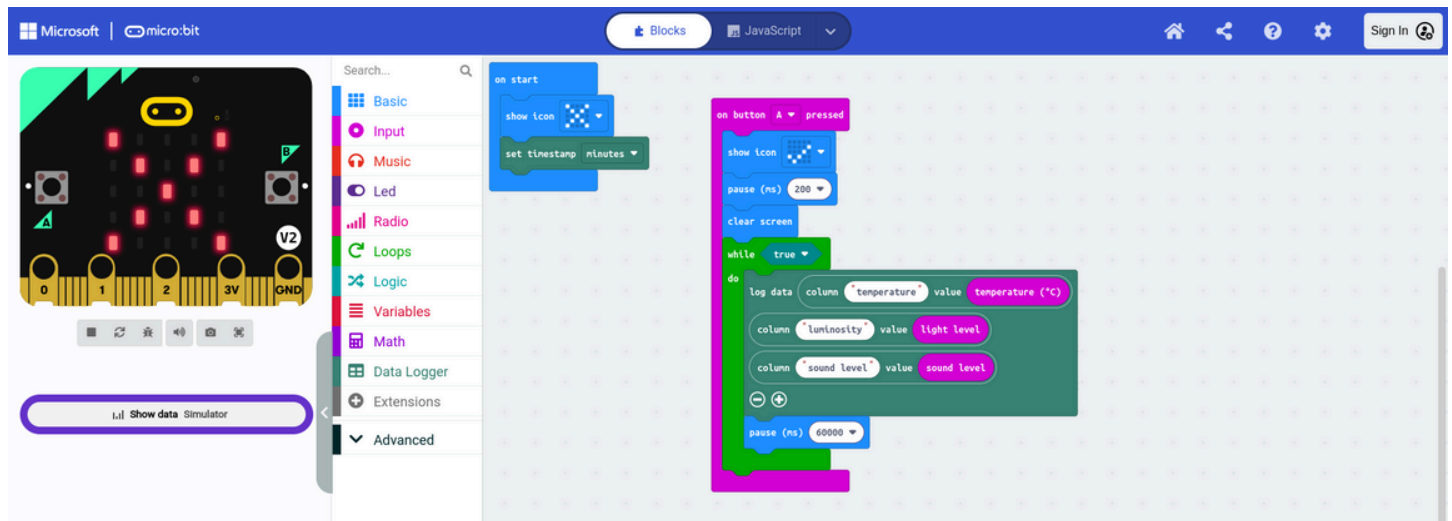
Cablaggio e utilizzo di una scheda Micro:bit

Segui questi passaggi per programmare, posizionare, registrare e recuperare dati ambientali utilizzando micro:bit.

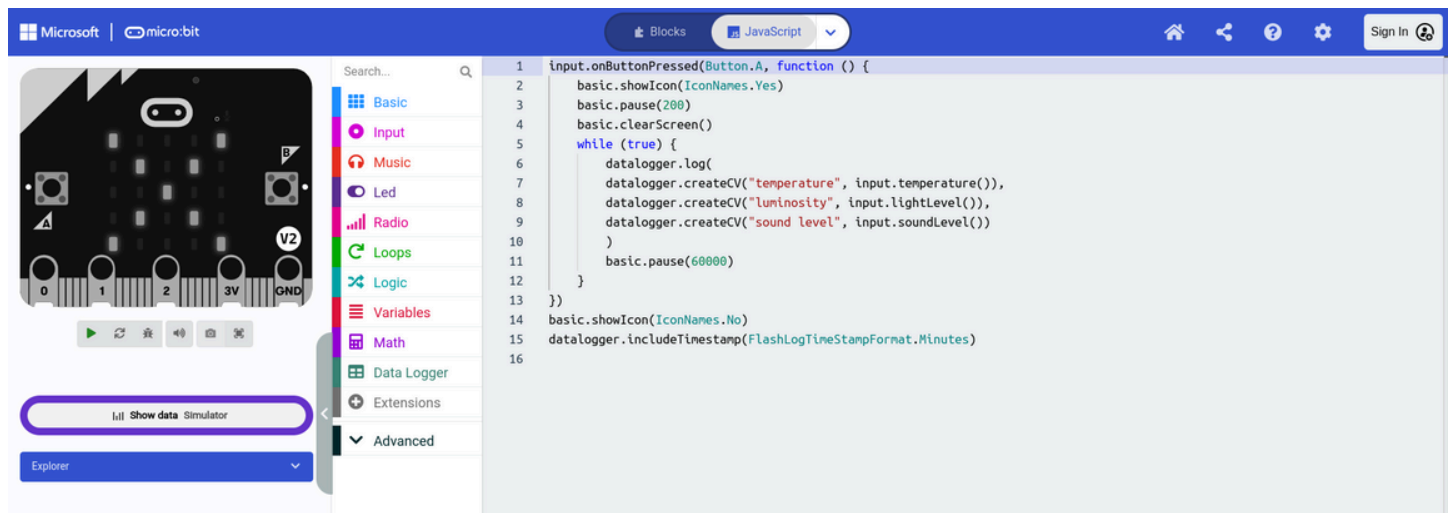
Fase 1 - Programmare la scheda Micro:bit

Collega la scheda Micro:bit: collega la scheda micro:bit al computer su cui hai creato il programma utilizzando l'editor MakeCode. Una volta collegata, la scheda micro:bit apparirà sul computer come un disco rimovibile (ad esempio, "MICROBIT"). Scrivi il programma: apri l'editor MakeCode per creare un programma che raccolga dati sul rumore utilizzando i sensori integrati nella scheda di programmazione Micro:bit V2. Assegna un nome chiaro al tuo progetto prima di iniziare. Una volta nell'editor, dopo aver creato il tuo nuovo progetto, verrai indirizzato alla schermata predefinita "out of the box" e dovrai installare un'estensione. Le estensioni in MakeCode sono gruppi di blocchi di codice che non sono direttamente inclusi nei blocchi di codice base di MakeCode. Le estensioni, come suggerisce il nome, aggiungono blocchi per funzionalità specifiche. Sono disponibili estensioni per un'ampia gamma di funzionalità molto utili, che aggiungono funzionalità per gamepad, tastiera, mouse, servocomandi e robotica e molto altro. Nelle colonne di visualizzazione dei blocchi, fai clic sul pulsante ESTENSIONI. Nell'elenco delle estensioni disponibili, trova l'estensione Datalogger che verrà utilizzata per questa attività. Fai clic sull'estensione che desideri utilizzare e nella schermata principale apparirà un nuovo gruppo di blocchi.

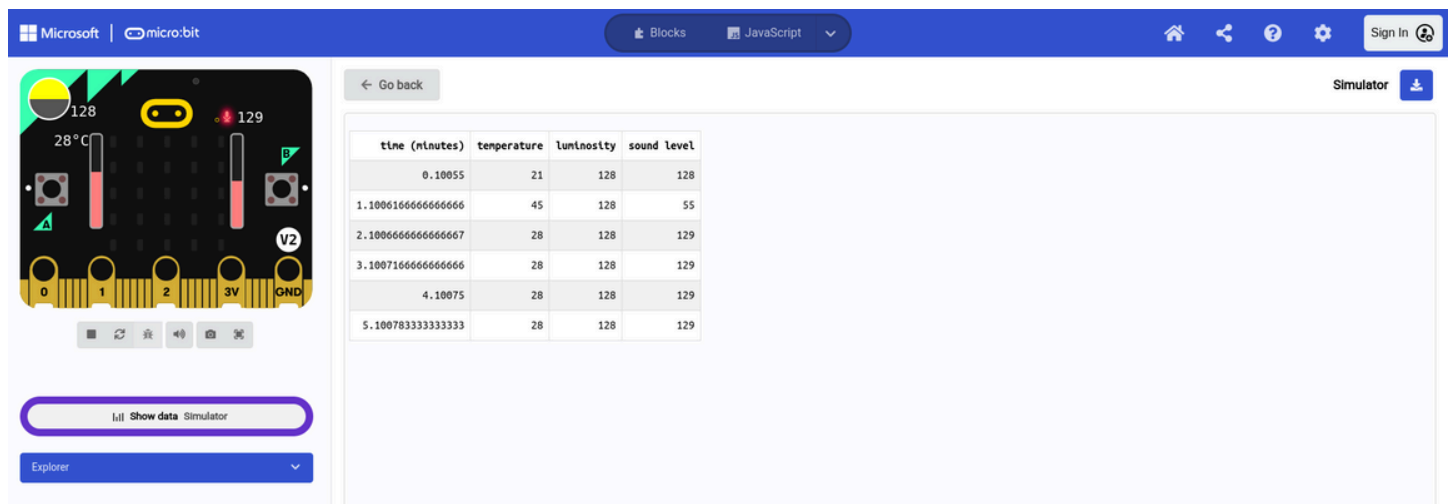
Quindi puoi iniziare a organizzare i tuoi blocchi seguendo il codice fornito di seguito (aggiungi un ciclo infinito, salva i dati nel datalogger...).



Puoi anche copiare e incollare il codice nell'editor Javascript.



Testare il programma utilizzando il simulatore MakeCode.



Una volta che il programma funziona correttamente sul simulatore, trasferiscilo sul tuo Micro:bit: clicca su "Carica" in MakeCode per generare un file .hex. Questo file contiene il programma compilato che permetterà alla scheda di funzionare.

Copia il file .hex dalla cartella dei download nell'unità rimovibile "MICROBIT".

Una volta copiato il file, la scheda si riavvierà automaticamente ed eseguirà il codice.

Fase 2: Posizionare il Micro:bit e iniziare a registrare i dati

Una volta programmato, posiziona il micro:bit per raccogliere i dati di cui hai bisogno, vicino al tuo letto, in un'area in cui possa registrare accuratamente luce, rumore e temperatura senza ostacoli. Utilizza un computer o un power bank per alimentare costantemente il micro:bit durante la registrazione. Assicurati che la scheda sia posizionata esattamente nella stessa posizione ogni notte per registrare dati confrontabili.

Prima di andare a letto, premere il pulsante "A" sul MicroBit per avviare la registrazione dei dati tramite il programma.

Fase 3: Recuperare i dati e preparare la scheda per la successiva sessione di registrazione

Ogni mattina, per evitare qualsiasi perdita di dati, consigliamo di scollegare il micro:bit dalla fonte di alimentazione per interrompere la registrazione dei dati e di collegare il micro:bit al computer per accedere al file compilato durante la notte dal datalogger (che si chiamerà "MY_DATA.HTM, disponibile sul lettore micro:bit).

Copia questo file sul tuo computer e rinominalo con la data odierna (ad esempio, BOARD1_NAME_YYYY-MM-DD.HTM).


Dopo aver copiato e rinominato il file, eliminare il file MY_DATA.HTM dalla scheda MicroBit per liberare spazio e consentire la registrazione di nuovi dati.

Ripetere il procedimento per la sessione successiva, ovvero la sera successiva prima di andare a letto.

Al termine del periodo di raccolta, sarà possibile recuperare tutti i file raccolti sulle diverse schede micro:bit. Una volta aperti, i file dati saranno accessibili in formato HTML. Forniranno tutti i dati raccolti e potranno essere scaricati in formato .csv.

Utilizzare e comprendere il codice

Ecco il codice JavaScript utilizzato per programmare una scheda micro:bit in modo che raccolga regolarmente dati su luce, rumore e temperatura:



```
input.onButtonPressed(Button.A, function () {
    basic.showIcon(IconNames.Yes)
    basic.pause(200)
    basic.clearScreen()
    while (true) {
        datalogger.log(
            datalogger.createCV("temperature", input.temperature()),
            datalogger.createCV("luminosite", input.lightLevel()),
            datalogger.createCV("niveau sonore", input.soundLevel())
        )
        basic.pause(60000)
    }
})
basic.showIcon(IconNames.No)
datalogger.includeTimestamp(FlashLogTimeStampFormat.Minutes)
```

Come funziona il codice?

Questo programma misura il rumore ambientale, la temperatura e la luminosità. Ogni minuto (l'intervallo può essere modificato a 10 secondi, 5 minuti, due volte all'ora, ecc.), il programma compila le informazioni in un data logger, da cui è possibile scaricare un file .csv.



Un file .csv (Comma-Separated Values) è un formato di file di testo utilizzato per archiviare dati tabulari (ad esempio in una tabella o un foglio di calcolo). Ogni riga del file rappresenta una riga di dati e ogni valore all'interno di una riga è separato da un delimitatore (spesso una virgola, ma a volte un punto e virgola o una tabulazione). È possibile recuperare i dati da un file .csv in un programma di fogli di calcolo come Excel o LibreOffice Calc. In Excel, apri il programma, fai clic su File > Apri, seleziona il file .csv e configura i delimitatori, se necessario, utilizzando lo strumento di importazione. In LibreOffice Calc, segui una procedura simile: fai clic su File > Apri, seleziona il file e utilizza la procedura guidata di importazione per scegliere il delimitatore (ad esempio, virgola o punto e virgola). In entrambi i casi, i dati vengono visualizzati in formato tabella, pronti per l'analisi.

Inizializzazione dell'evento di pressione del pulsante "A": quando l'utente preme il pulsante "A" sul MicroBit, viene attivata la funzione `input.onButtonPressed(Button.A, function () {...})`.

Visualizzazione dell'icona "Sì" durante l'esecuzione: prima di avviare la registrazione dei dati, il programma visualizza l'icona "Sì" (`basic.showIcon(IconNames.Yes)`) per 200 millisecondi (0,2 secondi) per indicare che il processo di registrazione è iniziato.

Pausa di 200 millisecondi: dopo aver visualizzato l'icona "Sì", il programma attende 200 millisecondi utilizzando `basic.pause(200)`.

Cancella schermo: dopo la pausa di 200 millisecondi, lo schermo viene cancellato con `basic.clearScreen()`, che lo prepara per ciò che segue senza essere intasato di immagini.

Ciclo infinito di raccolta dati: il programma entra in un ciclo `while` (vero) infinito. Ciò significa che i dati verranno raccolti e registrati all'infinito finché il MicroBit non verrà spento o riavviato.

Registrazione dei dati nel datalogger: ad ogni iterazione del ciclo, il programma registra i valori dei sensori MicroBit:

- temperatura: `input.temperature()`, che recupera la temperatura attuale in gradi Celsius.
- luminosità: `input.lightLevel()`, che misura il livello di luce ambientale.
- livello sonoro: `input.soundLevel()`, che cattura il livello sonoro ambientale.

Il livello sonoro e il livello di luminosità misurano un valore relativo e non hanno unità standard come i decibel (dB) per il livello sonoro o i lux (lx) per la luminosità. Più precisamente, il sensore misura l'intensità percepita. Questo valore è una stima numerica (da 0 a 255), dove 0 rappresenta il valore minimo (silenzio assoluto/buio totale) e 255 il valore massimo (rumore molto forte/luce intensa). La temperatura è misurata in gradi Celsius (°C). Questi valori vengono registrati nel datalogger come variabili con rispettivi nomi ("temperatura", "luminosità", "livello sonoro"). Questo viene fatto tramite la funzione `datalogger.log()`:

```
datalogger.log(  
datalogger.createCV("temperatura", input.temperature()),  
datalogger.createCV("luminosite", input.lightLevel()),  
datalogger.createCV("livello sonoro", input.soundLevel())  
)
```

La funzione `createCV` crea un "CV" (valore di contesto) per ciascun sensore e la funzione `datalogger.log` salva questi valori in un file sul MicroBit.

Pausa di 60.000 millisecondi prima della lettura successiva: dopo ogni registrazione, il programma attende 60.000 millisecondi (1 minuto) prima di leggere nuovamente i valori del sensore. Questo si ottiene con `basic.pause(60000)`. È possibile modificare la durata della pausa per acquisire più o meno dati (ad esempio, ogni minuto).

Timestamp dei dati (incluso tramite `datalogger.includeTimestamp`): oltre alla funzione relativa al pulsante, il comando `datalogger.includeTimestamp(FlashLogTimeStampFormat.Minutes)` viene utilizzato per includere un timestamp con

ogni record di dati. Il formato del timestamp è in minuti, il che significa che ogni record avrà un indicatore temporale basato sui minuti trascorsi dall'avvio del programma.

Visualizzazione dell'icona "No" prima dell'esecuzione: prima che l'utente prema il pulsante "A", il programma visualizza un'icona "No" (`basic.showIcon(IconNames.No)`) per indicare che il MicroBit è in attesa dell'azione dell'utente.