



Raccogli dati con il sensore di umidità e temperatura

Materiali e strumenti necessari

- Scheda Micro:bit V2 e sensori integrati: questa è la scheda principale programmabile. Include un sensore di luce (tramite il display LED), un sensore sonoro e un sensore di temperatura integrato. Prezzo consigliato: circa 19 € per scheda Micro:bit.
- Cavo micro-USB: consente di alimentare la scheda e programmarla da un computer.
- Batteria esterna (opzionale): utile per il funzionamento autonomo se la scheda è staccata dal computer. La batteria ufficiale Micro:bit è disponibile a circa 2,20 € l'una.



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.

- Sensore DHT22 (o DHT11): questi sensori sono molto diffusi per la misurazione di umidità e temperatura con microcontrollori. Il DHT11 è economico e sufficiente per progetti semplici, mentre il DHT22 offre una maggiore precisione e una risoluzione più elevata, a un costo leggermente superiore.
- Computer o tablet: utilizzato per scrivere il codice e trasferirlo al Micro:bit.
- Ambiente di programmazione: per una facile programmazione della scheda Micro:bit si consiglia l'editor online MakeCode.



Per questa fase, si consiglia di programmare da 3 a 6 schede Micro:bit, in modo da distribuirle tra gli studenti e raccogliere un volume di dati maggiore. È possibile svolgere l'attività con una sola scheda, ma ciò richiederà di prolungare il periodo di raccolta complessivo o di ridurre il tempo di raccolta per studente, da circa 7 giorni a 3 giorni.

Cablaggio e utilizzo di una scheda Micro:bit

Per programmare, installare, registrare e recuperare dati ambientali utilizzando una scheda Micro:bit, seguire i passaggi indicati di seguito.

Fase 1: Collegamento del sensore di temperatura/umidità alla scheda Micro:bit. Esistono due tipi di sensori DHT11/DHT22:

1. **Versione senza scheda PCB, con 4 pin;**
2. **Versione montata su PCB, con resistori pull-up integrati e solo 3 pin.**

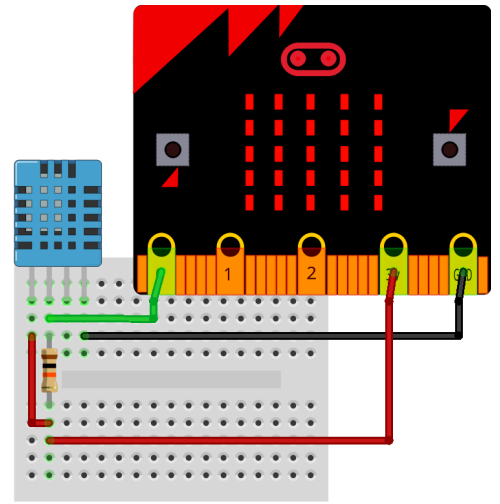
Si consiglia di utilizzare la versione PCB, più facile da collegare. Per la versione PCB (3 pin):

- **Vcc (+): collegare a 3,3 V o 5 V (entrambe le tensioni sono compatibili)**
- **GND (-): da collegare a terra (GND)**
- **Dati (OUT): connettersi a qualsiasi pin GPIO del Micro:bit**

Per la versione senza PCB (4 pin):

- È necessario aggiungere una resistenza di pull-up tra Vcc e Data per mantenere alto il pin Data. Una resistenza compresa tra 220 Ω e 10 k Ω funziona bene a 3,3 V; al di sopra di questa tensione, il sensore potrebbe non rispondere.
- È anche possibile utilizzare l'opzione pull-up interna del Micro:bit: in MakeCode, andare su "Pin" > "Altro" > "Imposta leva su pin...". Il Micro:bit ha resistori pull-up interni di circa 12–13 k Ω .
- **Nota: il terzo pin da sinistra (nella versione a 4 pin) non viene utilizzato.**

Collega il sensore DHT seguendo lo schema qui accanto.



Fase 2: Programmazione del Micro:bit.

Utilizzando il cavo USB, collega la scheda al computer tramite il connettore micro-USB. Una volta collegata, la scheda Micro:bit apparirà sul computer come unità rimovibile (ad esempio, "MICROBIT"). Apri l'editor MakeCode per creare un programma che raccolga dati di luce, rumore e temperatura utilizzando i sensori integrati di Micro:bit V2. Assegna un nome chiaro al tuo progetto prima di iniziare.

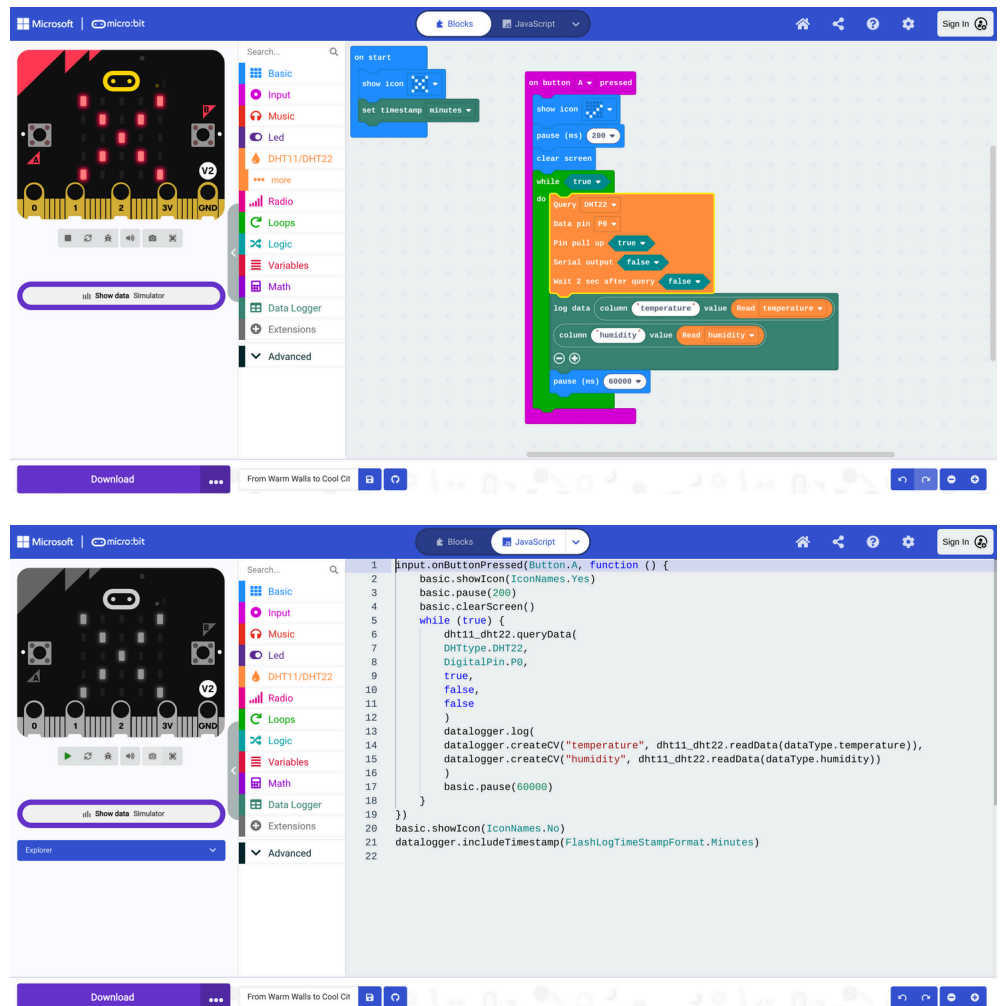
Una volta nell'editor e dopo aver creato il nuovo progetto, vedrai la schermata predefinita "pronto all'uso". Dovrai quindi installare un'estensione. Le estensioni in MakeCode sono gruppi di blocchi non direttamente inclusi nei blocchi base. Come suggerisce il nome, aggiungono blocchi per funzionalità specifiche. Esistono estensioni per un'ampia gamma di utilizzi: creare un gamepad, una tastiera, un mouse, controllare un servomotore, ecc. Nella colonna dei gruppi di blocchi, clicca su ESTENSIONI. Nell'elenco delle estensioni disponibili, trova l'estensione Datalogger, che verrà utilizzata per questa attività. Clicca sull'estensione desiderata: un nuovo gruppo di blocchi apparirà nella schermata principale. Fai lo stesso per il sensore di temperatura/umidità cercando l'estensione DHT11/DHT22.

Successivamente, puoi iniziare a organizzare i blocchi seguendo il codice fornito a lato (aggiungendo un ciclo infinito, salvando i dati nel datalogger, ecc.). È anche possibile copiare e incollare il codice direttamente nell'editor JavaScript.

Una volta che il programma funziona correttamente nel 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 3: Posizionare il Micro:bit e iniziare a registrare i dati

Una volta programmato, posiziona il Micro:bit in un luogo in cui possa misurare umidità e temperatura senza ostacoli per garantire letture affidabili. Utilizza un computer o un power bank per garantire l'alimentazione continua del Micro:bit durante la registrazione. Prima di andare a letto, premi il pulsante A sulla scheda Micro:bit per avviare la registrazione dei dati.

Fase 4: Recupero dei dati e preparazione della mappa per la sessione successiva

Ogni mattina, per evitare la perdita di dati, scollegare il Micro:bit dall'alimentazione per interrompere la registrazione. Quindi collegarlo al computer per accedere al file generato durante la notte dal datalogger. Questo file si chiama MY_DATA.HTM e si trova sull'unità MICROBIT.

- Copia questo file sul tuo computer.
- Rinominalo con la data odierna e un identificatore chiaro (ad esempio: BOARD1_NAME_YYYY-MM-DD.HTM).
- Una volta copiato e rinominato, eliminare il file MY_DATA.HTM dalla scheda Micro:bit per liberare spazio e consentire un nuovo salvataggio.

Ripeti questo processo ogni giorno per ogni carta utilizzata. Al termine del periodo di raccolta, puoi centralizzare tutti i file salvati su tutti i Micro:bit.

Utilizzare e comprendere il codice

Ecco il codice JavaScript utilizzato per programmare una scheda micro:bit per raccogliere regolarmente dati su umidità e temperatura:



```
input.onButtonPressed(Button.A, funzione () {  
  basic.showIcon(IconNames.Yes)  
  pausa di base (200)  
  basic.clearScreen()  
  mentre (vero) {  
    dht11_dht22.queryData(  
      Tipo DHT.DHT22,  
      DigitalPin.P0,  
      VERO,  
      falso,  
      falso  
    )  
    datalogger.log(  
      datalogger.createCV("temperatura",dht11_dht22.readData(dataType.temperatura)),  
      datalogger.createCV("umidita",dht11_dht22.readData(dataType.humidity))  
    )  
    pausa di base (60000)  
  }  
})  
basic.showIcon(IconNames.No)  
datalogger.includeTimestamp(FlashLogTimeStampFormat.Minutes)
```

Come funziona il programma?

Questo programma misura umidità e temperatura. A intervalli regolari – di default, ogni minuto, ma questa frequenza può essere modificata (ogni 10 secondi, ogni 5 minuti, due volte all'ora, ecc.) – il programma registra i dati in un datalogger, dal quale è possibile scaricare un file .csv.



Un file .csv (Comma-Separated Values) è un formato di file di testo utilizzato per archiviare dati tabulari, come quelli di una tabella o di un foglio di calcolo. Ogni riga del file corrisponde a una riga di dati e ogni valore è separato da un delimitatore, solitamente una virgola, ma a volte anche 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 software, clicca su File > Apri, seleziona il file .csv e quindi configura i delimitatori, se necessario, utilizzando lo strumento di importazione. In LibreOffice Calc, il processo è simile: clicca su File > Apri, seleziona il file e quindi utilizza la procedura guidata di importazione per impostare il delimitatore corretto (ad esempio, una virgola o un 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 eseguita 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)`.

Cancellazione dello schermo: dopo la pausa di 200 millisecondi, lo schermo viene cancellato con `basic.clearScreen()`, che lo prepara per ciò che segue senza riempirlo 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.

Query del sensore: i blocchi `dht11_dht22.queryData()` e `dht11_dht22.readData(...)` vengono utilizzati per selezionare il tipo di modulo e leggere i dati del sensore (si consiglia di rispettare un ritardo tra ogni query: almeno 1 secondo per il DHT11 e 2 secondi per il DHT22). È necessario effettuare una query preventiva per ottenere i valori di temperatura e umidità. Questo blocco verifica anche il checksum dei dati restituiti dal sensore. In caso di errore nel checksum, le letture di temperatura e umidità restituiranno -999 e il blocco "Ultima query riuscita?" restituirà false.

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

- Temperatura: il blocco `dht11_dht22.readData(dataType.temperature)` recupera la temperatura corrente in gradi Celsius.
- Umidità: il blocco `dht11_dht22.readData(dataType.humidity)` recupera l'umidità relativa corrente.

La temperatura si misura in gradi Celsius (°C) e l'umidità relativa in percentuale.

Questi valori vengono registrati nel datalogger come variabili denominate (rispettivamente "temperatura" e "umidità"). Questo avviene tramite la funzione `datalogger.log()`:

```
datalogger.log(  
datalogger.createCV("temperatura", dht11_dht22.readData(dataType.temperatura)),  
datalogger.createCV("umidita", dht11_dht22.readData(dataType.humidity))  
)
```

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 di rileggere: dopo ogni registrazione, il programma attende 60.000 millisecondi (1 minuto) prima di rileggere i valori del sensore. Questo si ottiene con `basic.pause(60000)`.

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.