# Multisensor-Temperatur

Schwierigkeitsgrad 🟡

## Benötigte Materialien

- 2 EMDACC4 mit Temperatur- und Feuchtigkeitssensor
- 1 EMDACC4 mit Magnetsensor
- 1 Magnet und etwas Klebeband, um ihn zu befestigen
- 1 Micro:bit pro Gruppe
- 1 Computer pro Gruppe

## Voraussetzungen

- Grundkenntnisse in Block-Sprachen
- Programmieren Sie die zentralen EMDACC4-Geräte mit der neuesten Firmware

## 1. Zentrale EMDACC4-Geräte bereitstellen

Zunächst muss jedes zentrale Gerät mit der neuesten Firmware programmiert werden.

Sobald dies geschehen ist, können die Micro:bits an ihren jeweiligen Platinen befestigt werden. Eine Platine wird im Klassenzimmer platziert, um die Innentemperatur zu messen, eine weitere außerhalb des Fensters, um die Außenbedingungen zu erfassen, und die dritte wird an den Magnetsensor angeschlossen, der überwacht, ob das Fenster geöffnet oder geschlossen ist. Nach dem Einschalten zeigt jedes Gerät ein Bestätigungssymbol auf dem Display an: „°C“ für die Temperaturplatinen und ein Schloss-Symbol für die Magnetplatine. Dies bedeutet, dass sie ordnungsgemäß angeschlossen sind und ihre Messwerte bereits über den Funkkanal übertragen.

Nun können die Sensoren befestigt werden: Der Außentemperatursensor sollte an einer Stelle angebracht werden, an der er vor direkter Sonneneinstrahlung und Regen geschützt ist, und das Magnetmodul sollte zusammen mit dem kleinen Magneten sicher am Fensterrahmen befestigt werden.

## 2. Verbinden Sie den Micro:bit mit dem Computer

Schalten Sie den Computer ein und öffnen Sie einen Webbrowser (Chrome wird empfohlen).

Gehen Sie zur MakeCode-Website (https://makecode.microbit.org/), erstellen Sie ein neues Projekt und geben Sie ihm einen aussagekräftigen Namen, zum Beispiel **„Multisensor-Daten**“.

Schließen Sie den Micro:bit an den USB-Anschluss des Computers an und koppeln Sie ihn mit dem MakeCode-Editor.

Schreiben Sie ein einfaches Programm zum Zeichnen von Diagrammen, um zu sehen, welche Werte die Sensoren übertragen.

Achten Sie auf die Namen der Geräte, denn diesmal müssen wir unterscheiden, von welchen Sensoren wir die Daten empfangen!

## 3. Entwerfen Sie den Algorithmus

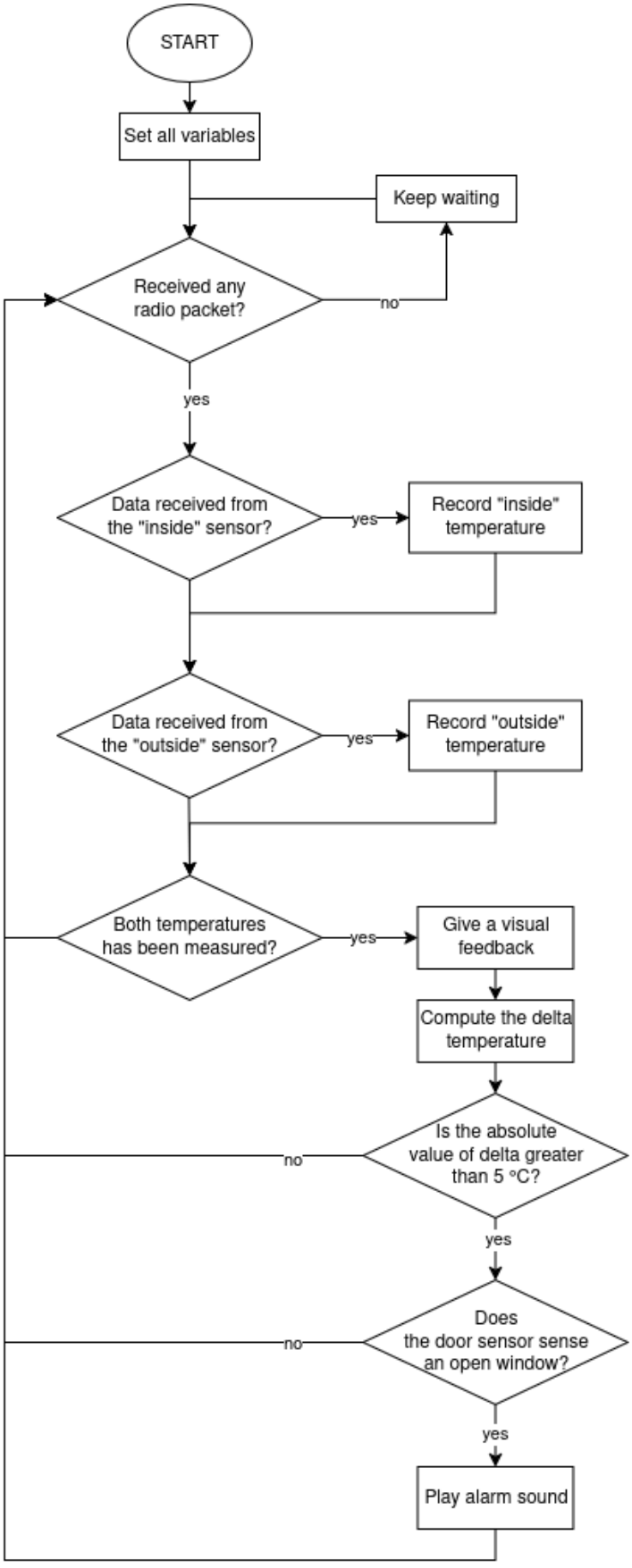
Die Idee hinter dieser Aktivität ist es, Benutzer zu warnen, wenn ein Fenster offen gelassen wird, während ein erheblicher Temperaturunterschied zwischen innen und außen besteht, da dies zu unnötiger Energieverschwendung führen würde.

Um dies zu erreichen, muss das auf dem Peripheriegerät Micro:bit ausgeführte Programm kontinuierlich Daten von den drei zentralen Geräten empfangen. Die aktuellen Werte der Innen- und Außentemperatur werden gespeichert und verglichen, um die Differenz zwischen ihnen zu berechnen.

Wenn das Fenster als offen gemeldet wird und der Temperaturunterschied größer als ein festgelegter Schwellenwert ist (z. B. 5 °C), sollte das Gerät einen Alarm auslösen. Wenn das Fenster geschlossen ist oder der Unterschied gering ist, ist keine Aktion erforderlich.

Diese Art der Argumentation spiegelt die reale Logik der Energieeinsparung wider und ist ein anschauliches Beispiel dafür, wie Sensoren kombiniert werden können, um nützliche Ergebnisse zu erzielen.

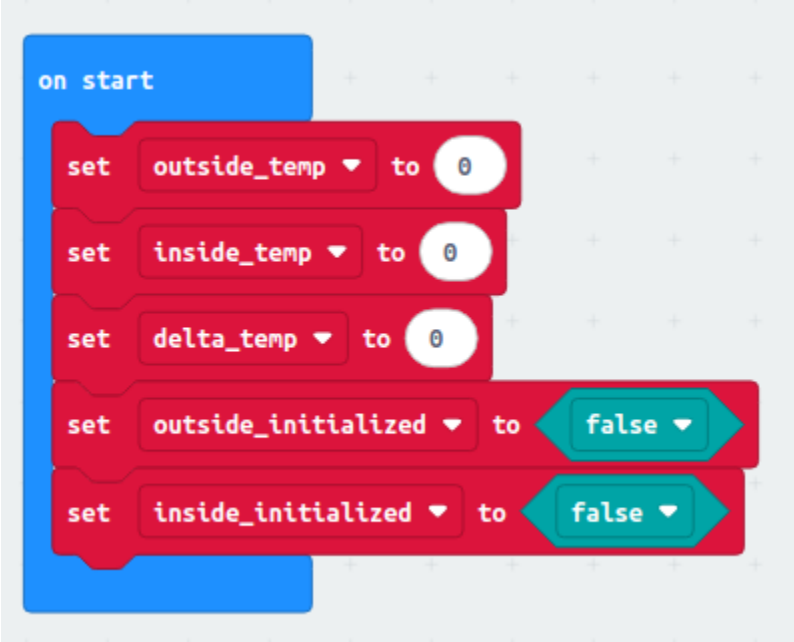
Wir können ein Flussdiagramm unseres gewünschten Verhaltens skizzieren:


START
Set all variables
Keep waiting
Received any radio packet?
no
yes
Data received from the "inside" sensor?
yes
Record "inside" temperature
Data received from the "outside" sensor?
yes
Record "outside" temperature
Both temperatures has been measured?
yes
Give a visual feedback
Compute the delta temperature
Is the absolute value of delta greater than 5 °C?
no
yes
Does the door sensor sense an open window?
no
yes
Play alarm sound

## 4. Programmieren Sie das Micro:bit

Jetzt ist es an der Zeit, mit der Erstellung des Codes zu beginnen. Zunächst einmal fällt auf, dass wir viele Variablen benötigen: eine für die Innen- und eine für die Außentemperatur, zwei weitere, um zu überprüfen, ob die Temperatur initialisiert wurde (um Berechnungsprobleme zu vermeiden, falls eine initialisiert wurde und die andere nicht), und natürlich die Variable zum Speichern der Temperaturdifferenz. Diese Art von Daten wird oft als „Delta" bezeichnet und ist in der Regel der absolute Wert der Differenz zwischen den beiden Werten. Sie wird verwendet, um zu messen, wie weit die Werte voneinander entfernt sind.

Zunächst können wir alle erforderlichen Variablen „bei Start" initialisieren:



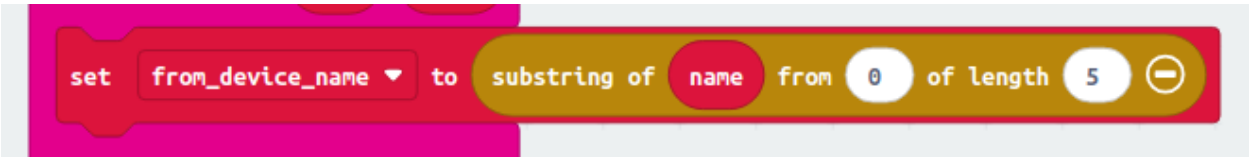

Der gesamte Code wird dann jedes Mal ausgeführt, wenn wir ein Datenpaket von einem unserer Sensoren empfangen, sodass wir den Block „bei Funkempfang <Name> <Wert>" benötigen.

Nun müssen wir eine Überlegung anstellen: Unser Code wird im Falle eines Temperaturpakets, eines Türstatuspakets, aber auch eines Feuchtigkeitspakets ausgeführt. Zusätzlich zu dieser Komplexität müssen wir die Tatsache berücksichtigen, dass es zwei Temperatursensoren geben wird, einen für den Innenbereich und einen für den Außenbereich.

Um diese neuen Probleme anzugehen, können wir ein einfaches Programm schreiben, um die Daten zu protokollieren, damit wir kontrollierte Experimente durchführen und den Namen der Quelle unseres Pakets verstehen können. Beachten Sie insbesondere, dass die Quellen mit 5 zufälligen Zeichen benannt sind, gefolgt von einem Unterstrich und einem Verweis auf den Sensortyp (TE im Fall von Temperatur).



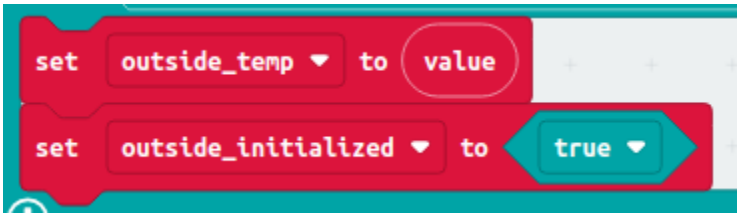

Sobald wir die 5-stelligen Namen des internen Sensors und der externen Sensoren haben, können wir mit der Programmierung unseres Algorithmus fortfahren. Es könnte sinnvoll sein, ihn in einer Variablen zu speichern, um etwas mehr Ordnung zu schaffen.




Jedes Mal, wenn wir ein Paket empfangen, müssen wir identifizieren, von welchem Gerät es stammt, und da wir die Namen abrufen konnten, müssen wir nun einen String-Vergleich (Erweitertes Menü > Text, nehmen Sie den Block „<dies> enthält <>“) innerhalb eines „if“-Blocks durchführen:

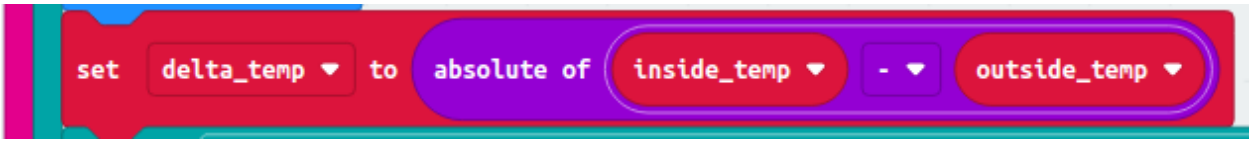



In jedem „if“-Block (einer pro Temperatursensor) weisen wir den gemessenen Wert der entsprechenden Variablen zu und setzen ihn als initialisiert:

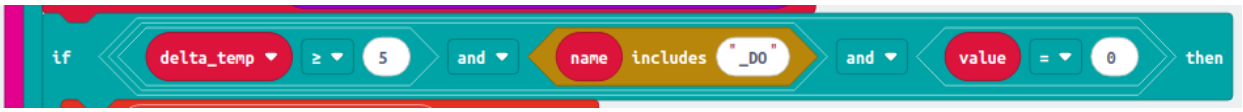


Wenn beide Sensoren korrekt initialisiert wurden, können wir ein visuelles Feedback geben, um zu signalisieren, dass das Gerät nun bereit ist, den Status des Fensters zu überwachen.

An dieser Stelle müssen wir die Formel zur Berechnung von „delta_temp“ wie oben beschrieben implementieren:




Anschließend benötigen wir einen „if“-Block, um zu überprüfen, ob wir über dem gewählten Schwellenwert (5 °C) liegen und ob das Fenster tatsächlich geöffnet ist (der Türsensor wird im Namen mit „DO” gekennzeichnet):




Wir müssen nur noch den Ton hinzufügen, der abgespielt wird, wenn die Bedingungen erfüllt sind, und schon sind wir fertig!

## Ideen

1. Ersetzen Sie den Ton durch eine visuelle Warnung (LEDs).
2. Protokollieren Sie „delta_temp“ im Laufe der Zeit über die serielle Schnittstelle und stellen Sie es grafisch dar, um Muster im Tagesverlauf zu erkennen.
3. Fügen Sie einen Timer hinzu: Lösen Sie den Alarm nur aus, wenn das Fenster länger als 1 Minute geöffnet ist und die Temperaturdifferenz hoch ist.
4. Untersuchen Sie Energieeinsparungen: Berechnen Sie den potenziellen Wärmeverlust, wenn das Fenster offen bleibt.

# Referenzcode

