Window Opening Detektor

Schwierigkeitsgrad 🟡

Benötigte Materialien

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

Voraussetzungen

- Fähigkeit, Daten mit der MakeCode-Plattform darzustellen
- Programmierung des zentralen EMDACC4 mit der neuesten Firmware

1. Einsatz des zentralen EMDACC4

Nachdem Sie die Firmware auf den Micro:bit geladen haben, der als zentrales Gerät fungieren wird, nehmen Sie das mit dem **Magnetsensor** ausgestattete EMDACC4-Modul und verbinden Sie den Micro:bit damit. Befestigen Sie die Batterie am Modul und schließen Sie das Gehäuse. Schalten Sie es ein und überprüfen Sie, ob das Symbol eines Schlosses auf dem Display erscheint: Dies bestätigt, dass die Platine richtig angeschlossen ist und bereits Magnetsensordaten über Funkwellen sendet.

Befestigen Sie nun das EMDACC4-Gerät und den Magneten an der Tür oder dem Fenster, das Sie überwachen möchten. Sehen Sie sich das Bild unten als Referenz an.

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 **„Open Window Detector”**.

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, um zu sehen, welche Werte der Sensor sendet, und beobachten Sie, wie sich die Werte je nach Anwesenheit des Magneten in der Nähe des Sensors ändern.

3. Entwerfen Sie den Algorithmus

Das Ziel dieser Aktivität ist es, das Peripheriegerät Micro:bit so zu programmieren, **dass** es **erkennt, wenn ein Fenster oder eine Tür zu lange offen steht**, und in diesem Fall einen Alarm auslöst. Dies könnte als Erinnerung daran dienen, Türen/Fenster zu schließen, um den Energieverbrauch für die Kühlung oder Beheizung eines Raumes zu reduzieren.

Beginnen wir damit, uns vorzustellen, wie das funktionieren soll. Am Ende des Prozesses sollten Sie etwas Ähnliches wie das folgende Flussdiagramm haben.

Machen Sie sich keine Sorgen, wenn es beim ersten Versuch nicht so klappt: Das Schreiben von Algorithmen erfolgt schrittweise. Man beginnt mit einer vagen Idee und verfeinert diese dann, indem man Details hinzufügt und die Sonderfälle überprüft. Stürzen Sie sich nicht zu schnell auf den Code, sonst könnten Sie etwas Wichtiges vergessen!

4. Programmieren Sie den Micro:bit

Nachdem Sie sich über den Algorithmus im Klaren sind, können Sie mit der eigentlichen Programmierung des Geräts fortfahren.

Alles beginnt mit dem Empfang eines Funkpakets vom zentralen EMDACC4-Gerät, sodass ein Block „on radio received <name> <value>” benötigt wird.

Immer wenn wir ein neues Paket vom zentralen Gerät empfangen, müssen wir eine Bedingung auswerten. Alle Blöcke, die mit Entscheidungen und Auswertungen zu tun haben, finden wir im Menü „Logik”. Insbesondere benötigen wir einen „if … then … else”-Block mit einer Vergleichsbedingung, die wir miteinander verbinden werden.

Die zu prüfende Bedingung ist, ob die Tür offen (Wert 0) oder geschlossen (Wert 1) ist. Um den Status der Tür und seine Änderungen zu verfolgen, müssen wir eine Variable deklarieren, die dazu dient, den Zeitpunkt aufzuzeichnen, zu dem wir benachrichtigt werden, dass die Tür geöffnet wurde. Wir können die Variable „opened_time“ über das Menü „Variables“ und durch Klicken auf „Make a Variable“ deklarieren.

Jedes Mal, wenn wir eine Variable erstellen, müssen wir einen Wert dafür festlegen. Es ist im Allgemeinen eine gute Idee, dies innerhalb des Blocks „on start” zu tun, um alles übersichtlich und unter Kontrolle zu halten.

Wenn wir „opened_time” auf 0 setzen, bedeutet dies so viel wie „die Tür ist geschlossen und es besteht keine Notwendigkeit, die verstrichene Zeit zu verfolgen”. Diese Anweisung wird jedes Mal wiederholt, wenn wir ein Paket erhalten, das uns mitteilt, dass die Tür geschlossen ist.

Wenn wir eine offene Tür erkennen, die zuvor als geschlossen gemessen wurde (eine weitere Bedingung), setzen wir „opened_time“ auf die aktuelle Zeit in Millisekunden (ms). Diesen Wert können wir aus dem Menü „Input – more“ (Eingabe – mehr) entnehmen. Auf diese Weise können wir die Zeit verfolgen, zu der wir zum ersten Mal gemessen haben, dass die Tür geöffnet wurde.

Wenn wir den Code wie im Flussdiagramm beschrieben zusammenstellen, kommen wir an einen Punkt, an dem wir messen müssen, wie viel Zeit seit der ersten Erkennung, dass die Tür offen gelassen wurde, vergangen ist. Da wir die Zeit gespeichert haben, zu der die Tür als offen gemessen wurde, müssen wir sie nur von der aktuellen Zeit subtrahieren, um die Anzahl der verstrichenen Millisekunden zu erhalten. Wenn wir einen Alarm auslösen möchten, wenn die Tür länger als 2 Minuten (120 Sekunden) offen steht, sollten wir einen ähnlichen Code schreiben:

Wenn die Bedingung erfüllt ist, können wir einen Ton für einen Takt abspielen, der als Alarm fungiert.

Nachdem Sie den Code doppelt überprüft haben, können Sie ihn auf den Micro:bit laden, um zu überprüfen, ob er wie erwartet funktioniert!

Ideen

1. Versuchen Sie, die Alarmreaktion zu ändern: Anstatt einen Ton abzuspielen, lassen Sie den Micro:bit blinken oder etwas anzeigen.
2. Kombinieren Sie dies mit anderen Daten: Was wäre, wenn das Gerät nur dann vor einem offenen Fenster warnt, wenn die Außentemperatur unter 18 °C liegt?

Referenzcode

