

Der Calliope bekommt ein Gedächtnis: Variablen

Modellvorstellung einer Variablen

Wir haben nun schon einige Aufgaben mit dem Calliope gelöst. Bei manchen wäre es hilfreich gewesen, wenn wir uns merken könnten, was vorher bereits passiert ist. So würde z. B. für den Lichtschalter eine Taste ausreichen, wenn wir dem Calliope Folgendes sagen könnten:

„Schalte beim Drücken der Taste A das Licht an, wenn es gerade aus ist.“

„Schalte beim Drücken der Taste A das Licht aus, wenn es gerade an ist.“

Da es keinen Block bzw. keinen Sensor gibt, der uns sagt, ob das Licht an oder aus ist, müssen wir uns merken, was wir zuletzt getan haben: das Licht an oder ausgeschaltet. Dazu benötigen wir eine Art Gedächtnis. Das sind beim Programmieren die *Variablen*.

Schauen wir uns erstmal etwas genauer an, was so eine Variable eigentlich ist. Wir können uns den Speicher des Rechners wie einen Schrank mit ganz vielen Schubladen vorstellen (s. Abbildung 1). In jede dieser Schubladen können wir ein kleines Zettelchen legen, auf dem wir notiert haben, was wir uns merken möchten. Eine Variable entspricht einer dieser Schubladen. Wenn wir eine neue Variable in unserem Programm anlegen, vergeben wir für diese Variable einen Namen. Damit reservieren wir uns eine der Speicherschubladen. Außen an die Schublade wird der Name der Variablen geschrieben, damit wir die Schublade wiederfinden. Für unseren Lichtschalter könnten wir z. B. eine Variable mit dem Namen `Licht_an` anlegen. In dieser Schublade legen wir dann den Wert *wahr* bzw. *falsch* ab, je nachdem, ob das Licht gerade an ist oder nicht. Denn das Schöne ist, dass wir den Wert in der Schublade auch verändern können.



Abbildung 1: Modellvorstellung eines Speichers

Erzeugen und Verwenden von Variablen im Programm

Die Blöcke zum Verwenden von Variablen befinden sich im Editor *MS Makecode*¹ in einem eigenen Bereich **Variablen**. Die Blöcke werden jedoch erst angezeigt, nachdem die erste Variable erstellt wurde.

Du erzeugst eine Variable, indem du auf die Schaltfläche *Erstelle eine Variable ...* klickst und dort den Namen deiner Variablen eingibst (s. Abbildung 3).

Nachdem du den Namen mit *OK* bestätigt hast, erscheint ein neuer Block mit dem Namen deiner Variablen (s. Abbildung 2).

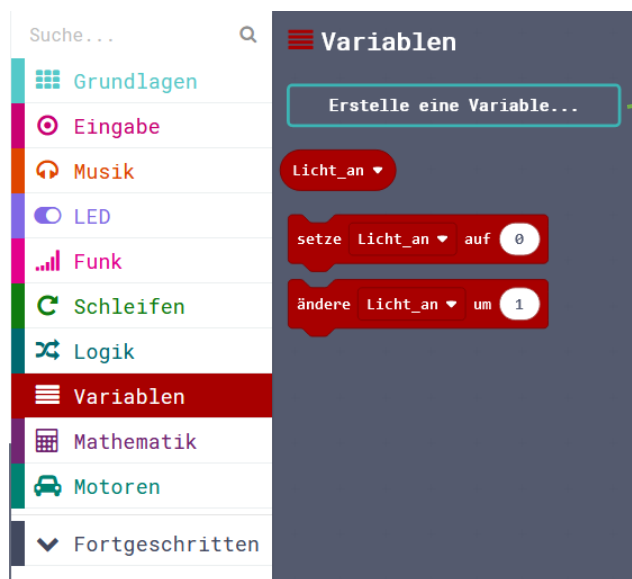


Abbildung 2: Blöcke für Variablen im Editor MS Makecode

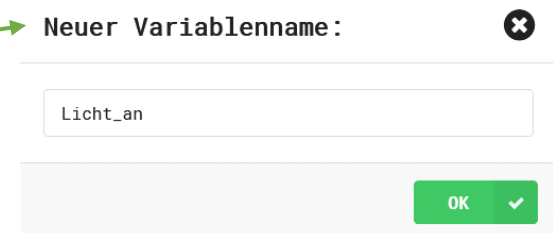


Abbildung 3: Erzeugen einer neuen Variablen

Zu Beginn enthält die Variable den Wert 0. Mithilfe der Blöcke *setze ... auf* bzw. *ändere ... um* kannst du nun den Wert der Variablen verändern. Der Wert kann eine Zahl, ein Text oder ein Wahrheitswert sein. Für unseren Lichtschalter ist ein Wahrheitswert passend, da wir nur zwei Zustände *wahr* (Licht ist an) und *falsch* (Licht ist aus) unterscheiden müssen.

Beim Starten des Programms legen wir zunächst den Wert *falsch* in der Variablen *Licht_an* ab, da die LED ja zu Beginn noch nicht leuchtet. In dem *setze ... auf*-Block, ersetzt du dazu die 0 durch den Block *falsch* (s. Beispiel 1).



Beispiel 1: Anfangswert der Variablen festlegen



Abbildung 4: Wertzuweisung für eine Variable im Schubladenmodell

¹ Es wird der Editor Microsoft MakeCode in der Version 3.0.33 verwendet: <https://makecode.calliope.cc/>

Die Ausführung des Blocks können wir uns so vorstellen, dass der Calliope in seinem Speicherschrank die Schublade mit der Beschriftung `Licht_an` heraussucht und einen Zettel mit der Aufschrift *falsch* hineinlegt.

Wenn das erste Mal die Taste A gedrückt wird, sollten nun die Anweisungen in Beispiel 2 ausgeführt werden.

Die LED wird eingeschaltet und der Calliope merkt sich in der Variablen `Licht_an` mit dem Wert *wahr*, dass das Licht jetzt an ist. Der Wert in der Variablen `Licht_an` muss also geändert werden. Das können wir uns auch so vorstellen: Der Zettel in der Schublade `Licht_an` wird herausgenommen.

Der Wert *falsch*, der auf dem Zettel steht, wird wegradiert und stattdessen der Wert *wahr* auf den Zettel geschrieben. Anschließend wird der Zettel wieder in die Schublade gelegt.



Beispiel 2: Ändern des Wertes einer Variablen

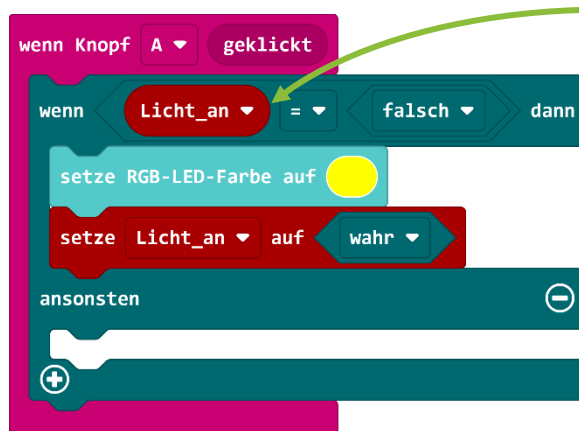


Abbildung 6: Wertänderung einer Variablen im Schubladenmodell

Variablen auslesen

In unserem Programm können wir den Wert, der in der Variablen gespeichert ist, auch abfragen, ohne ihn zu verändern. Jedes Mal, wenn in unserem Programm der Block mit dem Namen der Variablen auftaucht, schaut der Calliope in der entsprechenden Schublade nach und setzt den Wert, den er dort findet, anstelle der Variablen in das Programm ein. Die Variable steht also für den aktuellen Wert, der in der entsprechenden Schublade liegt.

Wir verwenden nun den Wert in unserer Variablen `Licht_an` für unseren Lichtschalter und erweitern das Programm aus Beispiel 2:



Beispiel 3: Simulation eines Lichtschalters mit Taste A

Wenn der Knopf A gedrückt wird, macht der Calliope Folgendes: Für den Variablen-Block `Licht_an` wird der Wert eingesetzt, der aktuell in der entsprechenden Schublade gespeichert ist. Wenn dies der Wert *falsch* ist, dann ist die Bedingung „*Licht_an = falsch*“ erfüllt und es wird der *dann*-Zweig ausgeführt. Das Licht wird eingeschaltet und in der Variablen `Licht_an` wird nun der Wert *wahr* gespeichert. Dadurch wäre die Bedingung „*Licht_an = falsch*“ beim nächsten Mal nicht mehr erfüllt und es würde der *ansonsten*-Zweig ausgeführt. Hast du schon eine Idee, welche Blöcke in den *ansonsten*-Zweig eingesetzt werden müssen? Du findest es bestimmt heraus!

Merke: Mithilfe einer **Variablen** kann man den Calliope dazu bringen **auf die gleiche Eingabe** (Taste A gedrückt) **unterschiedlich zu reagieren** (LED einschalten oder LED ausschalten), je nachdem in welchem **Zustand** sich der Calliope gerade befindet.

Aufgabe 1: Als du den Umgang mit den Sensoren des Calliope geübt hast, hast du eine Alarmanlage für Freddy's Federmappe erstellt. Zur Erinnerung schauen wir uns noch einmal die Aufgabenstellung an:

Freddy ärgert sich, dass sein Nachbar manchmal an seine Federmappe geht und Stifte herausnimmt, ohne ihn zu fragen. Er beschließt daher mithilfe des Calliope eine kleine Alarmanlage zu bauen. Den Calliope legt er dazu in seine Federmappe. Sobald sie geöffnet wird, soll der Alarm losgehen.

- a) Wenn du das Programm abgespeichert hast, öffne es und schaue es dir noch einmal genau an. Ansonsten erstelle ein entsprechendes Programm für den Calliope und teste es, indem du den Calliope in deine Federmappe legst.

Freddy findet es ein wenig nervig, dass die Alarmanlage die ganze Zeit piept, wenn er selbst einen Stift aus der Federmappe holen will.

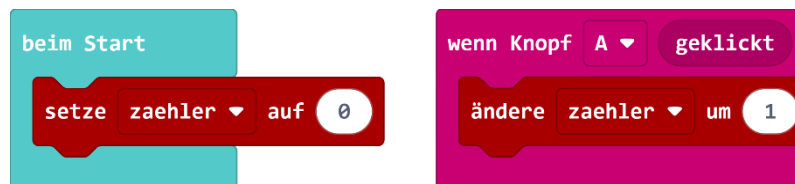
- b) Ändere dein Programm so, dass Freddy die Alarmanlage durch Drücken der Taste A deaktivieren und durch Drücken der Taste B wieder scharf schalten (aktivieren) kann.
- c) Versuche dein Programm so zu ändern, dass du nur die Taste A benötigst. Bei jedem Druck auf die Taste A soll zwischen den Zuständen *aktiviert* und *deaktiviert* gewechselt werden.
- d) Fallen dir noch andere Programme ein, die du bereits erstellt hast und bei denen eine Aktivierung und Deaktivierung sinnvoll wäre oder bei denen eine Unterscheidung verschiedener Zustände hilfreich wäre? Verwende Variablen, um diese Programme zu optimieren.

Variablen als Zahlenspeicher

Wenn wir in einer Variablen Zahlen speichern, können wir den Wert der Variablen auch mithilfe des **ändere ... um** -Blocks verändern.

Aufgabe 2:

- a) Wie verhält sich der Calliope, wenn das Programm in Beispiel 4 ausgeführt wird? Wenn du keine Idee hast, probiere es aus. Nimm dazu zunächst die Ergänzung aus Aufgabenteil b) (1) vor.



Beispiel 4: Anwendung des **ändere...um**-Blocks

- b) Erweitere das Programm aus Beispiel 4 so, dass
- (1) der aktuelle Wert der Variablen auf dem Display ausgegeben wird.
 - (2) der Wert der Variablen mithilfe der Taste B auf den Wert 0 zurückgesetzt werden kann.
- c) Verändere das Programm aus Beispiel 4 so, dass in Fünfer-Schritten gezählt wird.
- d) Fallen dir sinnvolle Anwendungen für das Programm aus a) ein? Wie müsste das Programm dafür ggf. angepasst werden?

Aufgabe 3: In Hamburg gibt es nicht nur Busse und Straßenbahnen, sondern auch Schiffe, die im Hamburger Hafen einige Verbindungen abdecken. Diese sind vor allem bei Touristen sehr beliebt, so dass sie regelmäßig überfüllt sind und den überzähligen Passagieren der Zutritt verweigert werden muss. Mehr als 400 Personen darf ein Schiff nicht befördern.

Konstruiere mithilfe deines Calliope eine Lichtschranke, die jeden Passagier zählt, der die Lichtschranke passiert. Mithilfe der Tasten A und B soll zwischen einsteigenden und aussteigenden Passagieren unterschieden werden können. Die LED zeigt an, ob noch Platz an Bord ist oder die zulässige Passagierzahl erreicht wurde. Vielleicht signalisiert eine dritte Farbe, dass das Schiff bald ausgelastet ist oder die aktuelle Anzahl an Passagieren wird auf dem Display ausgegeben.

Tipp 1: Um die Lichtschranke zu konstruieren benötigst du eine Taschenlampe, die du auf den Lichtsensor richtest. Anhand der gemessenen Lichtstärke erkennst du, ob der Lichtstrahl unterbrochen wird.

Tipp 2: Zum Testen kannst du auch erstmal davon ausgehen, dass das Schiff bereits mit 4 Passagieren voll besetzt ist.

Aufgabe 4: Im Bereich **Mathematik** gibt es Blöcke zum Rechnen und zum Erzeugen von Zufallszahlen.

- a) Verwende diese Blöcke, um dir einen digitalen Würfel zu konstruieren, der die gewürfelte Seite auf der LED-Matrix ausgibt.
- b) Erstelle verschiedene Varianten des Würfels, z. B. einen Farbwürfel, einen akustischen Würfel für Blinde oder einen Würfel der nur gerade Zahlen würfelt.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Von der Lizenz ausgenommen ist das InfSI-Logo.

Für die korrekte Ausführbarkeit der Quelltexte in diesem Leitfaden wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

Bildnachweis: Die Fotos wurden vom Autor selbst erstellt. Die Screenshots wurden mithilfe des online-Editors *MS Makecode* in der Version 4.0.29 erstellt.