

14

Helligkeitssensor – „Wiederhole-bis-Schleife“ („while-Schleife“)

Darum geht's

Über das LED-Display kann der Calliope messen, wie hell es ist. Den Messwert wandelt er in Zahlen zwischen 0 und 255 um. 0 ist dabei der kleinste Wert. Dieser Wert sagt aus, dass es absolut dunkel ist. 255 ist der größte Wert und bedeutet, dass es ganz hell ist. Mithilfe dieses Sensors kann man verschiedene Programme schreiben, z. B. für eine Taschenlampe, die ab einem bestimmten Wert automatisch angeht.

Zudem ist der Calliope mit einer RGB-LED ausgestattet. Die Abkürzung „RGB“ steht für Rot-Grün-Blau. Aus diesen drei Farben lassen sich alle anderen Farben mischen, so wie beispielsweise beim Malen mit Wasserfarben. Mischt man etwa Gelb und Blau, entsteht Grün. Welcher Farbton entsteht, hängt davon ab, wie viel man von der jeweiligen Grundfarbe dazugibt. Im Zusammenhang mit dem Calliope wird die „Menge“ der Farbe mithilfe von Zahlen zwischen 0 und 255 angegeben. 0 bedeutet: Nimm nichts von dieser Farbe. 255 bedeutet: Nimm sehr viel von dieser Farbe.

Das wird programmiert

Die Kinder programmieren eine Taschenlampe. Bei einer bestimmten Dunkelheit schaltet sich das LED-Licht automatisch ein. Für die Programmierung benötigen die Kinder die „Während-wahr-“ bzw. „Wiederhole-bis-Schleife“ („while-Schleife“). Alles, was innerhalb der Schleife steht, wird so lange wiederholt, bis die Bedingung nicht mehr erfüllt ist. Eingesetzt wird diese Schleife, wenn die Anzahl der Wiederholungen unbekannt ist.

Kompetenzerwartungen

Die Kinder lernen

- einen weiteren Sensor des Calliope kennen: den Helligkeitssensor.
- die „Während-wahr-“ bzw. „Wiederhole-bis-Schleife“ („while-Schleife“) anzuwenden.
- das RGB-Licht kennen. Sie verstehen, wie sie verschiedene Farben aus Rot, Grün und Blau mischen können.

Das bereiten Sie vor

- Kopieren Sie die Programmieranleitung „Die automatische Taschenlampe“ für jedes Kind. (S. 95–97)

Stundenverlauf**Einstieg (5 Minuten)**

Lesen Sie folgende Geschichte vor: „Tim ist heute bei seinem besten Freund Paul eingeladen. Sie basteln gerade an einem Dinosaurier, als Paul Tim fragt: „Sag mal, hast du die Schere gesehen? Ich kann sie nirgends finden.“ Die beiden Freunde suchen überall: Auf dem Fußboden, in der Spielzeugkiste und unter den Büchern. Auch im Schrank und hinter dem Regal schauen Tim und Paul nach, doch dort ist es zu dunkel. „Hast du vielleicht eine Taschenlampe? Ohne kann ich nichts erkennen“, meint Tim. „Nein, aber wir können uns ja eine programmieren,“ antwortet Paul. „Ja? Dann aber eine, die selbstständig das Licht ein- und ausschaltet. Das haben doch schon einige Autos. Bei denen schaltet sich bei Dunkelheit die Beleuchtung auch von selbst an.“

Besprechen Sie die Geschichte zunächst kurz anhand der folgenden Fragen:

- Gibt es Verständnisfragen?
- Was ist passiert?
- Haben die Kinder eine Vermutung, wie das Programm für eine automatische Taschenlampe aussehen könnte?

14. Helligkeitssensor – „Wiederhole-bis-Schleife“ („while-Schleife“)

Arbeitsphase (30 Minuten)

Teilen Sie die Programmieranleitung „Die automatische Taschenlampe“ aus. Lassen Sie die Kinder anschließend – wenn möglich – selbst die Aufgaben lösen. Klären Sie während der Arbeitsphase die Fragen der Schülerinnen und Schüler und/oder unterstützen Sie sie aktiv bei der Durchführung des Projekts.

Abschluss/Reflexion (10 Minuten)

Die Kinder versammeln sich im Sitzkreis. Lassen Sie eines der Kinder das Programm und den dazugehörigen Code vorführen. Wissen die Kinder, welcher Code was bewirkt? Wie müsste man den Code für die Anregungen ganz am Ende der Programmieranleitung abändern?

Lösung:

```

wenn Knopf A gedrückt
  während
    wahr
      wenn (Lichtstärke < 100)
        dann
          setze LED-Farbe auf
            Rot: 248
            Grün: 248
            Blau: 255
            Weiß: 0
  
```

```

wenn Knopf A gedrückt
  während
    wahr
      wenn (Lichtstärke < 100)
        dann
          spiele Note Mittleres C für 1 Takt
  
```

Die automatische Taschenlampe (1/3)

Diesmal erstellst du einen Code, mit dessen Hilfe dein Calliope zur Taschenlampe wird. Wenn es dunkel wird, schaltet sie sich automatisch ein.

Öffne den Editor „Makecode“ und gib deinem neuen Projekt einen Namen (Beispiel: Taschenlampe).

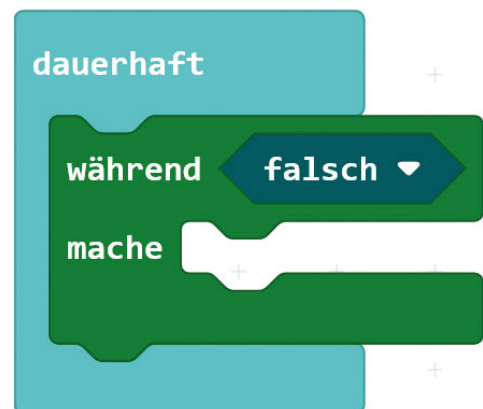
Lösche den Befehl „beim Start“.

Du brauchst den Befehl „dauerhaft“. Damit überprüft dein Calliope immer wieder, wie hell es ist.


Suche nun die Anweisung  und ziehe sie in in den Block „dauerhaft“.

Du findest sie in der Gruppe „Schleifen“.

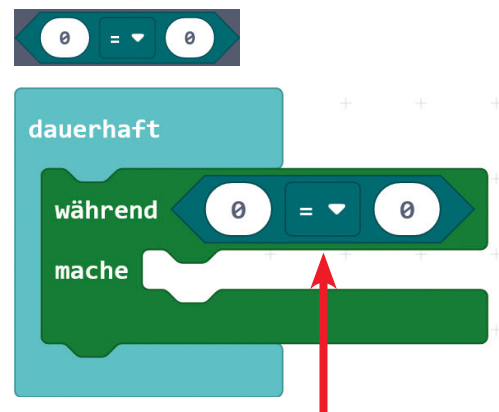
Die Anweisung ist eine Schleife. Sie sorgt dafür, dass der Calliope wieder und wieder überprüft, wie hell es ist.



Nun muss der Calliope noch wissen, was er machen soll. Beispielsweise „während“ der Helligkeitssensor Dunkelheit feststellt, „dann“ soll ein Licht angehen.

Hierfür benötigst du folgenden  -Block. Du findest ihn unter .

Füge den  -Block hinter „während“ anstelle von „falsch“ ein.

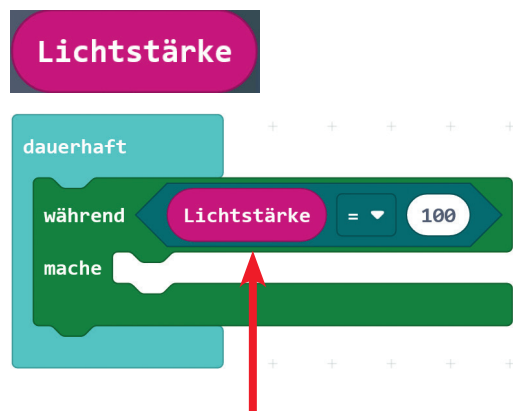


Ersetze die erste 0 von  mit dem Block .

Diesen findest du unter .

Nun kannst du die Lichtstärke messen. **Ersetze die zweite 0 durch deine Helligkeitsgrenze.**

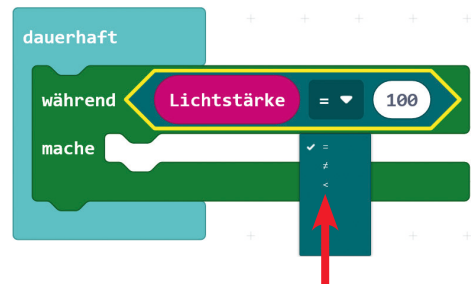
Dies ist eine Zahl zwischen 0 (totale Finsternis) und 255 (ganz hell). Versuche es mal mit 100.



Die automatische Taschenlampe (2/3)

Hinter dem Block **Lichtstärke** und vor der 100 steht bisher ein „=“. Allerdings soll die LED-Lampe angehen, wenn es dunkler ist als 100 – und nicht nur wenn die Helligkeit einen Wert von 100 hat.

Klicke auf das kleine Dreieck und wähle „<“.

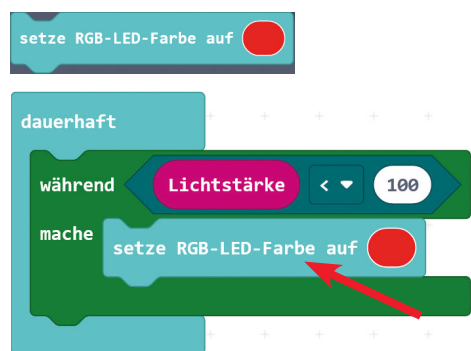


Wenn der Helligkeitssensor einen Dunkelheitswert von weniger als 100 misst, soll etwas passieren – die LED soll angehen.

Suche den Block  .


Du findest ihn unter  Grundlagen .

Füge den Block nach „dann“ ein.

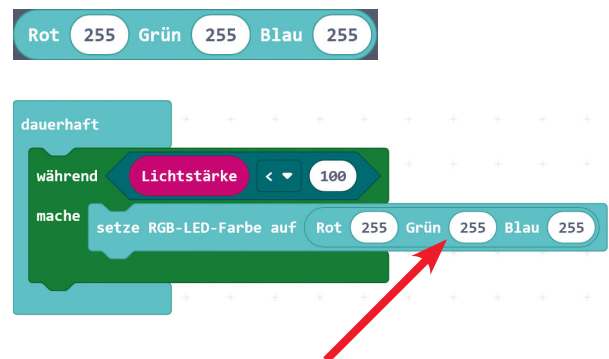


Die Taschenlampe soll jedoch nicht rot, sondern weiß leuchten.

Dafür benötigst du den Block  .

Du findest ihn unter der Gruppe  Grundlagen .

Füge ihn bei  **anstelle der roten Kachel ein.**



Die Anweisung ist für die RGB-LED.

RGB bedeutet Rot-Grün-Blau. Man kann alle Farben aus diesen 3 Farben mischen.

Diese Farben kannst du mithilfe von Zahlen zwischen 0 und 255 einstellen.

Für Weiß benötigst du folgende Werte:

- Rot: 248
- Grün: 248
- Blau: 255

Die automatische Taschenlampe (3/3)

Wenn Calliope eine Helligkeit von 100 oder mehr misst, soll die LED-Lampe wieder ausgehen.

Hierfür benötigst du den Befehl . Diesen findest du unter mehr. Er erscheint, wenn du anklickst. Füge den Befehl unter der -Anweisung ein.

Du kannst dir auch die Lichtstärke erst einmal anzeigen lassen und später den Wert in der -Anweisung abändern.

Dafür benötigst du den Befehl . Diesen findest du in der Gruppe .

Platziere den Befehl im -Block, jedoch außerhalb der -Anweisung.

Einen zweiten -Befehl fügst du direkt in der -Anweisung unter dem Block ein.

Anschließend fügst du bei beiden -Befehlen bei der 0 den -Befehl ein.

Wenn du den Befehl nicht mehr benötigst, kannst du ihn später auch wieder herauslöchen.

Schließe deinen Calliope über das USB-Kabel an deinen Computer an und übertrage das Programm.

Nimm den Calliope vom Computer und verbinde ihn mit dem Batteriehalter.

Probiere auch Folgendes aus:

- Ändere das Programm so ab, dass es sich erst dann einschaltet, wenn du zum Beispiel auf Knopf A drückst.
- Baue eine Alarmanlage. Wenn der Helligkeitssensor Dunkelheit misst, soll Calliope einen Alarmton von sich geben. Du kannst dann den Calliope an einem Versteck anbringen. Wird ein Schatten, zum Beispiel von einem Erwachsenen, auf den Lichtsensor geworfen, gibt dein Calliope Alarm.

19

Kommunikation
untereinander**Darum geht's**

Calliope-Boards können auch untereinander kommunizieren. Das funktioniert ähnlich wie beim Radio. Ein Calliope sendet mithilfe von Funkwellen ein Signal aus (zur Info: Der Calliope kann auch über Bluetooth kommunizieren). Alle anderen Calliope, die die gleiche Frequenz eingestellt haben, können das Signal empfangen. Ähnlich wie man beim Radio den Sender wechseln kann, funktioniert das auch bei den Calliope-Boards. Die Calliopes können Gruppen zugeteilt werden. Dann hören sie nur die Nachrichten der jeweiligen Gruppe. Einstellen kann man diese Gruppe mit dem Befehl „Setze Gruppe“ aus der Gruppe „Funk“ und bei diesem unter „Mehr“.

Das wird programmiert

Die Kinder erstellen ein Programm, mit dessen Hilfe ein Calliope andere, versteckte Calliopes aufspüren kann.

Kompetenzerwartungen

Die Kinder verstehen, dass mehrere Calliopes Nachrichten untereinander austauschen können, ohne dass sie mit einem Kabel verbunden oder an einen PC angeschlossen sein müssen.

Das bereiten Sie vor

- Kopieren Sie die Programmieranleitungen „Das Versteckspiel“ (für die, die suchen, und für die, die sich verstecken) 1-mal für zwei 2er-Gruppen. (S. 127 – 133)

Stundenverlauf**Einstieg (7 Minuten)**

Fragen Sie die Kinder, wie sie bisher dem Calliope Anweisungen – also die selbst geschriebenen Programme – übertragen haben. Erklären Sie ihnen anschließend den Inhalt von „Darum geht's“. Beantworten Sie etwaige Fragen.

Arbeitsphase (30 Minuten)

Teilen Sie die Kinder in 2er- oder 3er-Gruppen ein. Weisen Sie nun den Gruppen verschiedene Gruppennummern zu. Achten Sie darauf, dass immer 2 Gruppen die gleiche Nummer erhalten. Diese Gruppennummer

sollen die Kinder in das weiße Feld mit der 0 der „Setze Gruppe“-Anweisung eintragen. Teilen Sie anschließend die Programmieranleitung aus. Die Gruppe, die die anderen sucht, erhält die Anweisung „Das Versteckspiel – Code für den Calliope, der sucht“. Die andere erhält die Anleitung „Das Versteckspiel – Code für den Calliope, der sich versteckt“. Lassen Sie die Kinder anschließend – wenn möglich – selbst programmieren. Klären Sie während der Arbeitsphase die Fragen der Schüler und/oder unterstützen Sie sie aktiv bei der Durchführung des Projekts.

Abschluss/Reflexion (8 Minuten)

Ihre Schüler versammeln sich im Sitzkreis.


Fragen Sie die Schüler:

- Wie funktioniert das Versteckspiel?
- Wie funktioniert die Kommunikation der Calliope untereinander?
- Warum habt ihr unterschiedliche Zahlen beim Befehl „setze Gruppe“ eingegeben?
- Was passiert, wenn alle Calliope in der gleichen Gruppe wären?
- Kennt ihr ein Beispiel aus dem Alltag? (Funkgerät)

Hinweis

Die Signalstärke des versteckten Calliopes kann von der im folgenden angegebenen Signalstärke abweichen. Dann funktioniert die Anzeige auf dem suchenden Calliope nicht so, wie es beschrieben ist. Hier muss zuerst die Signalstärke ermittelt werden. Dafür wird benötigt man folgenden Code:

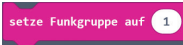




Dieser wird auf den suchenden Calliope übertragen. Lassen Sie nun die Kinder mit dem suchenden Calliope langsam auf den versteckten Calliope zugehen und sich die Signalstärke aufschreiben. Wie hoch ist diese, wenn die Kinder weit weg sind bzw. wie hoch ist diese, wenn beide Calliope aufeinanderliegen? Überlegen Sie nun, ab welcher Entfernung Calliope die unterschiedlich großen Kreise anzeigen soll. Tragen Sie diese Werte in das zweite weiße Feld in die Befehle  ein (Punkt 9 und 10 vom „Code für den Calliope, der sucht“).

Das Versteckspiel (1/4) – Code für den Calliope, der sich versteckt

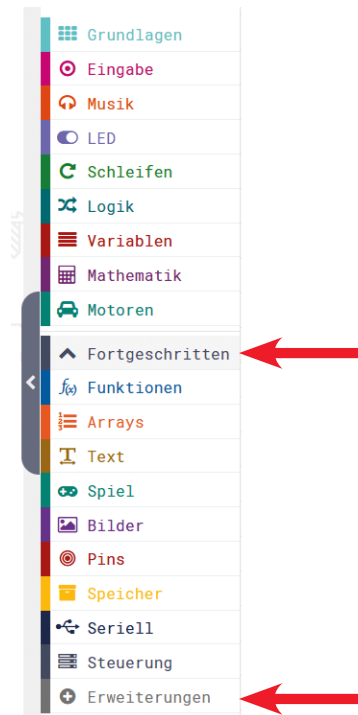
2 Calliopes sollen miteinander „sprechen“. Damit das geht, müssen sie sich über eine Funkfrequenz unterhalten.

Teile deinem Calliope mit, auf welcher Funkfrequenz er sich mit dem anderen Calliope unterhalten soll.


Dazu benötigst du den Befehl  aus der Gruppe .


Die -Blöcke musst du zuerst in die Bibliothek laden. Klicke hierfür zuerst auf „Fortgeschritten“ und dann auf „Erweiterungen“.

Den Befehl findest du unter der Gruppe und dort unter „Mehr“.



Du siehst folgendes Bild.

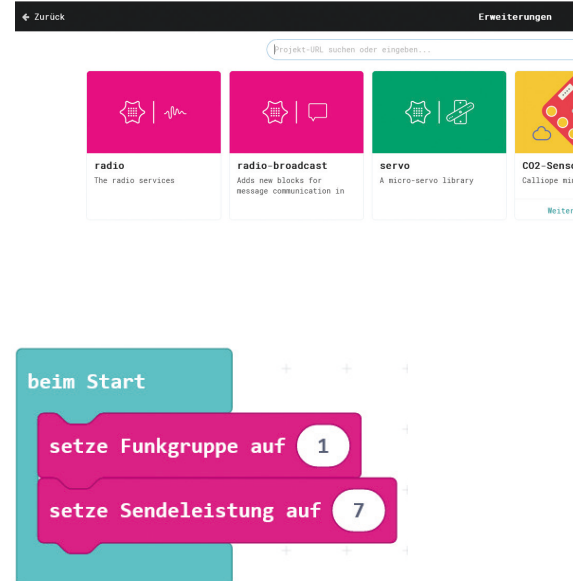
Suche  **und klicke darauf.**

Die Gruppe „Funk“ steht nun direkt unter .

Du benötigst die Befehle „beim Start“, „setze Funkgruppe auf 1“ und „setze Sendeleistung auf 7“.

„Setze Sendeleistung auf 7“ findest du unter „Funk“ und dort unter „Mehr“.

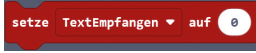
In kleineren Räumen solltest du die Zahl von 7 auf zum Beispiel 4 ändern. Dein Calliope funkt dann schwächer und kann so nicht so schnell gefunden werden.



Das Versteckspiel (2/4) –

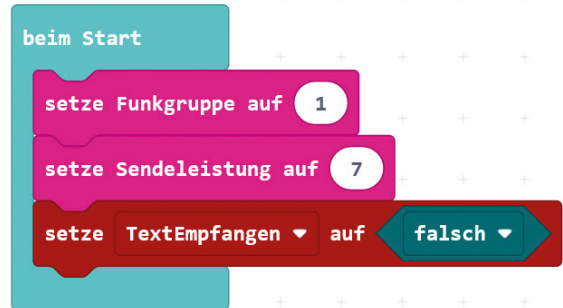
Code für den Calliope, der sich versteckt

Erstelle eine neue Variable mit dem Namen „TextEmpfangen“.

Platziere den Code  im Befehl „beim Start“ unterhalb von „setze Sendeleistung auf 7“.


Ersetze die 0 durch „falsch“ aus der „Logik“-Gruppe.

Mit diesem Code sagst du deinem Calliope, dass er Texte, die er aus früheren Spiele empfangen hat, löschen soll.



Platziere die Anweisungen so, wie du sie rechts sehen kannst.

Nun musst du deinen Calliope dazu bringen, zu funkeln – und zwar dauerhaft, damit er von einem anderen Calliope gefunden werden kann.

Dafür benötigst du den -Befehl.




Platziere darin die Anweisung .

Schreibe in das weiße Kästchen beispielsweise „Ich bin hier“.

Was du genau schreibst, ist egal. Wichtig ist nur, dass dein Calliope etwas funkt.

Damit du auch siehst, dass dein Calliope wirklich funkt, kannst du dir zum Beispiel einen blinkenden Smiley auf den LEDs anzeigen lassen.

Allerdings soll dein Calliope nur funkeln, „wenn“ er keinen Text von dem suchenden Calliope empfangen hat. Dafür brauchst du

,  aus dem Bereich „Logik“ sowie  aus dem Bereich „Variablen“.

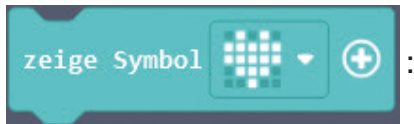


Das Versteckspiel (3/4) –

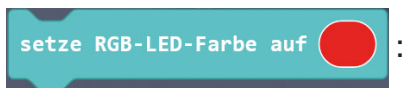
Code für den Calliope, der sich versteckt

Dein Calliope soll anzeigen, wenn der andere Calliope ihn gefunden hat.

Dafür brauchst du folgende Anweisungen:



Ändere das Bild in einen Haken.

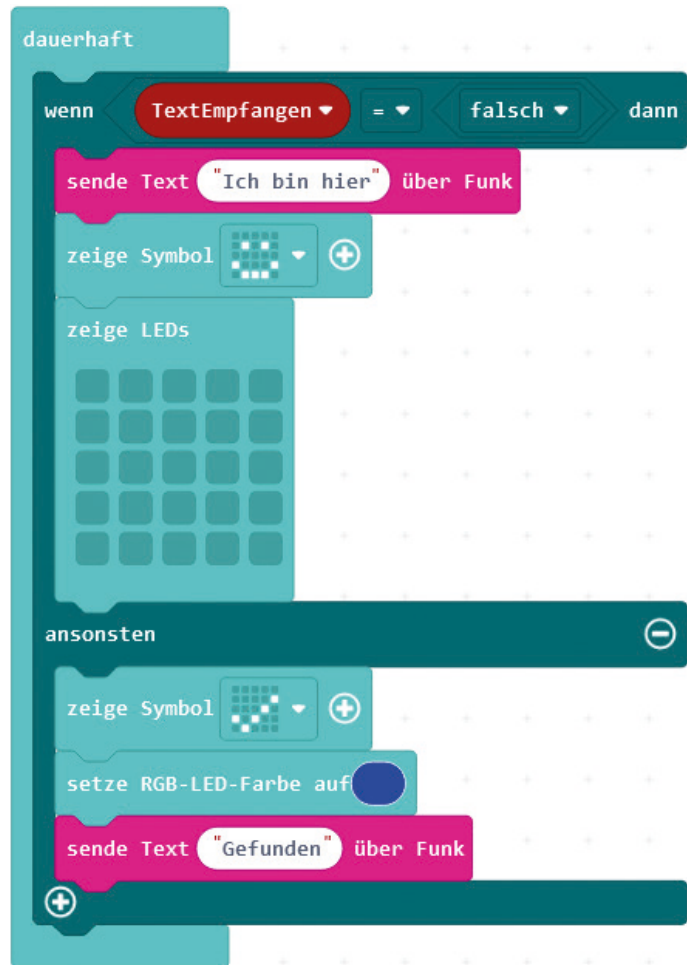


Wähle die Farbe Blau.



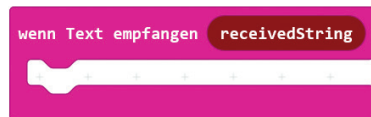
aus „Funk“:


Hier kannst du „gefunden“ in das Kästchen schreiben.



Wenn dein Calliope gefunden wurde, sendet der suchende Calliope einen Text. Diesen Text muss ein Calliope noch empfangen können.

Dafür benötigst du den Befehl



„Received String“ ist englisch und bedeutet in diesem Fall „empfangener Text“. Du findest den Befehl in der Gruppe  „Funk“.

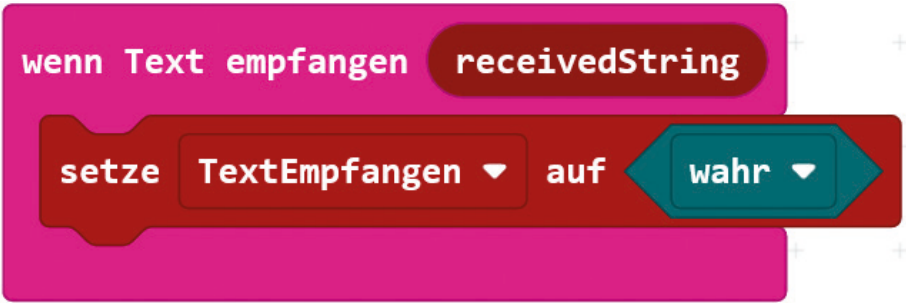
Wenn dein Calliope den Text empfängt, muss auch etwas passieren.

Dafür brauchst du  aus der Gruppe „Variablen“ und

 aus „Logik“.

Das Versteckspiel (4/4) –

Code für den Calliope, der sich versteckt



wenn Text empfangen `receivedString`

setze `TextEmpfangen` auf `wahr`

Schließe deinen Calliope über das USB-Kabel an deinen Computer an, speichere das Programm und übertrage es.

Nimm den Calliope vom Computer und verbinde ihn mit dem Batteriehalter.

Starte dein Programm, verstecke dich und lasse dich nun von deinem Partner suchen.

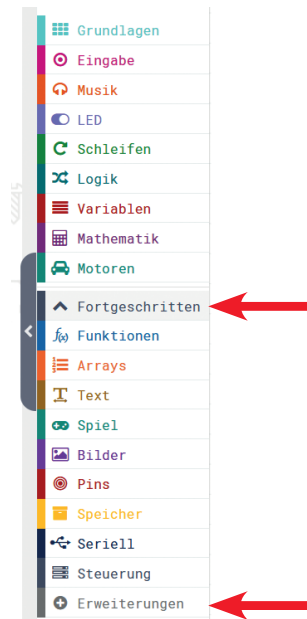
Das Versteckspiel (1/3) – Code für den Calliope, der sucht

Teile deinem Calliope mit, auf welcher Funkfrequenz er sich mit dem anderen Calliope unterhalten soll.

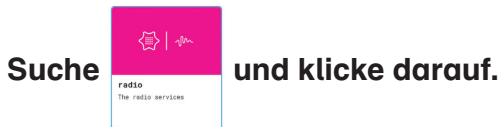
Dazu benötigst du den Befehl **setze Funkgruppe auf 1** aus der Gruppe **Funk**.

Die **Funk**-Blöcke musst du zuerst in die Bibliothek laden. Klicke hierfür zuerst auf „Fortgeschritten“ und dann auf „Erweiterungen“.

Den Befehl findest du unter der Gruppe und dort unter „Mehr“.



Du siehst folgendes Bild.

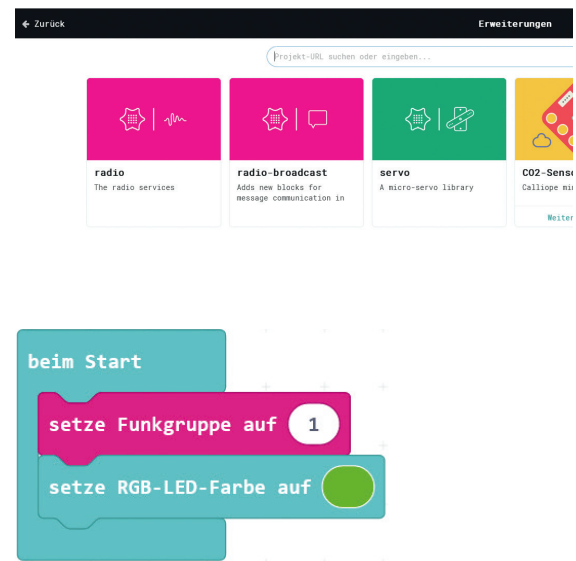


Die Gruppe „Funk“ steht nun direkt unter **LED**.

Du benötigst die Befehle „beim Start“, „setze Funkgruppe auf 1“ und „setze RGB-LED-Farbe auf rot“.

Ändere die RGB-LED-Farbe von „Rot“ auf „Grün“.

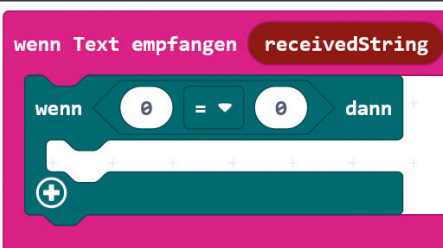
Nun siehst du, wenn dein Calliope eingeschaltet ist und sucht.



Damit dein Calliope den anderen Calliope auch sucht, benötigst du den Befehl **wenn Text empfangen receivedString**.

Du findest den Befehl in der Gruppe **Funk**. „Received String“ ist englisch und bedeutet „empfangener Text“.

Dein Calliope soll prüfen, wie stark das Signal ist, das der andere Calliope aussendet. Nur so weiß er, wie weit der andere noch weg ist. Dafür brauchst du die Anweisungen „wenn wahr dann“ und „0=0“ aus der Gruppe „Logik“.



Das Versteckspiel (2/3) – Code für den Calliope, der sucht

Der versteckte Calliope sendet ein Signal aus. Wenn dein Calliope weit von dem versteckten Calliope weg ist, ist das Signal sehr schwach. Je näher du kommst, desto stärker wird das Signal.



Um zu entscheiden, ob die Signalstärke stark oder schwach ist, brauchst du den Befehl „Empfangenes Paket Signalstärke“.

Platziere diesen in dem ersten weißen Feld mit der 0 von „0=0“, ändere das „=“ in „<“ und schreibe in das zweite weiße Feld anstelle der 0 eine -60.

```

wenn Text empfangen receivedString
  wenn Empfangenes Paket Signalstärke < -60 dann
  
```

Wenn du den versteckten Calliope gefunden hast, soll bei deinem Calliope die mittlere LED leuchten. Die LED-Farbe soll von Grün auf Blau wechseln.

Suche dazu die Befehle  und  aus der Gruppe

[Grundlagen](#)

Der versteckte Calliope zeigt an, dass er gefunden wurde, wenn dein Calliope ihm dies sagt.

Suche dafür den Block

 aus der Gruppe

[Funk](#)

Schreibe als Text zum Beispiel „Gefunden“ in das weiße Feld.

```

wenn Text empfangen receivedString
  wenn Empfangenes Paket Signalstärke < -60 dann
    zeige LEDs
    setze RGB-LED-Farbe auf
    sende Text "Gefunden" über Funk
  sonst wenn dann
  sonst wenn dann
  ansonsten
  
```

Eigentlich bist du nun schon mit deinem Programm fertig. Du kannst es aber noch etwas verbessern und dir beispielsweise die Signalstärke durch unterschiedlich große Kreise anzeigen lassen.

Klicke dafür 3-mal auf das „+“ in dem Block „wenn wahr dann“.

Das Signal des versteckten Calliopes ist umso stärker, je näher dein Calliope ihm kommt. Mit folgenden Anweisungen sagst du deinem Calliope, was er machen soll, wenn er vom versteckten Calliope etwas weiter weg ist oder wenn er ganz weit weg ist.

Das Versteckspiel (3/3) – Code für den Calliope, der sucht

Füge in den ersten „wenn sonst“-Code folgende Befehle ein:

- **und**: Platziere ihn direkt hinter „sonst wenn“ von „wenn wahr dann“.
- **2-mal** **=**: Füge sie in die beiden Felder von **und** ein.
- **2-mal** **Empfangenes Paket Signalstärke**: Platziere ihn in das erste weiße Feld von **=** anstelle der 0.
- **Ändere das „=“ des ersten** **=** in „<“ und die 0 in -60.
- **Ändere das „=“ des zweiten** **=** in „≥“ und die 0 in -70.
- **setze RGB-LED-Farbe auf**: Stelle Hellblau ein.
- **zeige LEDs**: Klicke die mittleren LEDs an, so wie du es unten sehen kannst.

Nun musst du den Code für die zweite „wenn sonst“-Anweisung erstellen.

Wiederhole dafür die vorherigen Befehle, ändere die Codes jedoch so ab, wie du es rechts sehen kannst.

Füge ganz unten in „wenn wahr dann“ Folgendes ein:

- „zeige Text „hi!““
- „Empfangenes Paket Signalstärke“:
- **Füge diese Anweisung in das weiße Feld von „zeige Text „hi!““ ein.**
- „setze RGB-LED-Farbe auf rot“

```

wenn Text empfangen receivedString
  wenn Empfangenes Paket Signalstärke > -60 dann
    zeige LEDs
    setze RGB-LED-Farbe auf Hellblau
    sende Text "Gefunden" über Funk
  sonst wenn Empfangenes Paket Signalstärke <= -60 und Empfangenes Paket Signalstärke < -70 dann
    zeige LEDs
    setze RGB-LED-Farbe auf Hellblau
  sonst wenn Empfangenes Paket Signalstärke <= -70 und Empfangenes Paket Signalstärke <= -100 dann
    zeige LEDs
    setze RGB-LED-Farbe auf Hellblau
  ansonsten
    zeige Text Empfangenes Paket Signalstärke
    setze RGB-LED-Farbe auf rot
  
```

Schließe deinen Calliope über das USB-Kabel an deinen Computer an, speichere das Programm und übertrage es.

Nimm den Calliope vom Computer und verbinde ihn mit dem Batteriehalter.

Starte dein Programm und suche deinen Partner.



Spiel programmieren – Array

Darum geht's

Im Spiel, das programmiert werden soll, leuchten vier LEDs. Die X- und Y-Werte für diese LEDs soll sich Calliope „merken“, was bedeutet, sie müssen gespeichert werden. Würde man Variablen verwenden, müsste man 4 verschiedene Variablen für die X- und 4 verschiedene Variablen für die Y-Werte anlegen. Einfacher geht dies mit Listen oder – wie es unter Programmierern heißt – „Arrays“. Man kann sich diese Arrays ähnlich einer Einkaufsliste vorstellen: Angenommen, man möchte 3 Dinge kaufen. Dann erstellt man eine Einkaufsliste mit 3 Positionen. Anschließend schreibt man die Dinge, die benötigt werden, an die einzelnen Positionen, z. B.:

1. Banane
2. Äpfel
3. Brot

Unterschieden werden müssen jedoch die Position und der Inhalt. Was ist z. B. die Antwort auf die Frage, was an 2. Position steht? (Äpfel). Es kann aber auch gefragt werden: „Stehen Äpfel auf der Liste?“ In diesem Fall geht es nicht um die Positionsnummer, sondern um das, was in der Position steht. Genauso verhält es sich auch mit Arrays.

Das wird programmiert

In dem Spiel geht es darum, 4 leuchtende LEDs so schnell wie möglich zu fangen.

Kompetenzerwartungen

Die Kinder

- verstehen komplexe Programme und können diese erstellen.
- kennen die Unterschiede zwischen den verschiedenen Schleifen und Bedingungen und können sie entsprechend verwenden.
- lernen Arrays kennen und diese einzusetzen.

Das bereiten Sie vor

- Kopieren Sie die Programmieranleitung „Wer fängt alle leuchtenden LEDs?“ 1-mal für jede 2er-Gruppe. (S. 136–144)
- Erstellen Sie das Programm und übertragen Sie es auf 2 Calliope-Boards.

Hinweis

Einfacher ist es, wenn Sie vorher bereits Stunde 18 und 19 durchgeführt haben. Sie können das Erstellen des Codes auch auf zwei Stunden á 45 Minuten aufteilen.

20. Spiel programmieren – Array

Stundenverlauf

Einstieg (5 Minuten)

Erzählen Sie den Kindern, dass sie heute ein Spiel programmieren sollen. Ziel des Spiels ist es, als Erster alle leuchtenden LEDs zu fangen. Schalten Sie die beiden Boards ein und führen Sie das Spiel vor. Überlegen Sie gemeinsam, wie ein Code hierfür aussehen könnte (nur skizzieren, nicht genau ausformulieren). Sagen Sie den Kindern, dass sie bisher Variablen kennengelernt haben. Für das Spiel sind diese jedoch ungeeignet. Man benötigt Arrays. Erläutern Sie Ihren Schülern mithilfe des Textes „Darum geht’s“, was Arrays sind.

Arbeitsphase (35 Minuten)

Teilen Sie die Kinder in 2er- oder 3er-Gruppen ein. Alle Kinder der Klasse werden später gegeneinander spielen. Wenn nur die Kinder einer 2er- oder 3er-Gruppe gegeneinander spielen sollen, müssen Sie Gruppennummern vergeben (siehe Kapitel „Kommunikation untereinander“) und den Block „Setze Gruppe“ zusätzlich in den Code einbauen. Teilen Sie die Anleitung aus. Lassen Sie die Kinder anschließend – wenn möglich – selbst programmieren. Klären Sie während der Arbeitsphase die Fragen der Schüler und/oder unterstützen Sie sie aktiv bei der Durchführung des Projekts.

Abschluss/Reflexion (5 Minuten)

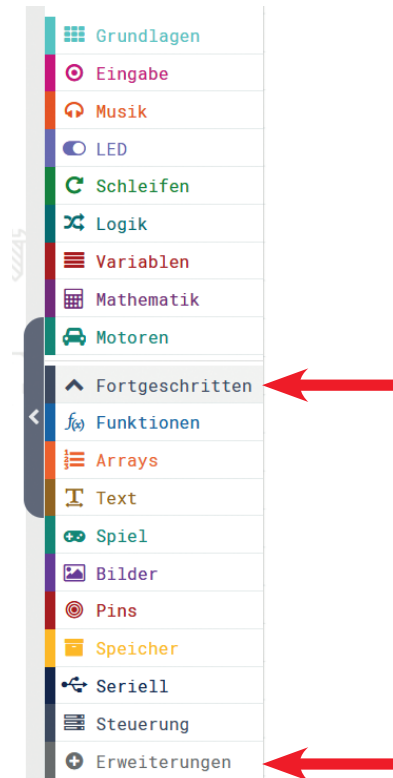
Ihre Schüler versammeln sich im Sitzkreis. Besprechen Sie gemeinsam den Code für das Spiel. Wo gab es Probleme, was haben die Kinder nicht verstanden? Wie könnte man das Programm verbessern? Sind Probleme beim Spiel aufgetreten? Beispielsweise kann es vorkommen, dass ein Calliope zufällig 2 starre, leuchtende LEDs an derselben Stelle findet. Dann leuchten beim Spielstart nur 3 anstatt 4 LEDs. Oder der bewegliche Punkt befindet sich am Anfang des Spiels an derselben Stelle, an der sich auch der starre Punkt (starre leuchtende LED) befindet. Wie könnte man diese Probleme beheben?

Wer fängt alle leuchtenden LEDs? (1/9)


Beim Start soll dein Calliope 4 Lampen an einer zufälligen Stelle auf dem LED-Display aufleuchten lassen. Diese roten Punkte sind starr und bewegen sich nicht. Zudem soll an einer bestimmten Stelle eine weitere LED aufleuchten. Dieser Punkt soll sich bewegen, wenn du das Board hin und her bewegst. Ziel des Spiels ist es, mit dem beweglichen Punkt so schnell wie möglich alle starren Punkte „einzusammeln“.

Lade zuerst die  **Funk** -Blöcke in die Bibliothek.

Klicke hierfür zuerst auf „Fortgeschritten“ und dann ganz unten auf die „Erweiterungen“.

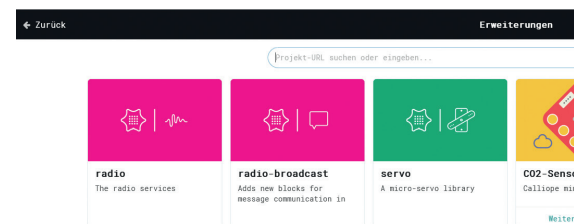


Du siehst folgendes Bild.

Suche  und klicke darauf.

Natürlich könntest du die Werte für jede einzelne Lampe bestimmen. Diesen müsstest du allen einen ähnlichen, aber dennoch leicht unterschiedlichen Namen geben, zum Beispiel „ZufallspunktX1“, „ZufallspunktX2“, „ZufallspunktX3“, „ZufallspunktX4“.

Das ist jedoch sehr umständlich.



Wer fängt alle leuchtenden LEDs? (2/9)

Um es dir leichter zu machen, benötigst du ein Array.
Darin werden die Werte, die dein Calliope zufällig erzeugt hat, gespeichert.

Lege zuerst 2 neue Variablen mit den Namen „ArrayX“ und ArrayY“ an.

Nun benötigst du die „beim Start“-Anweisung.

Klicke im Menü auf „Fortgeschritten“ und anschließend auf „Arrays“.

Suche den Befehl „setze list auf Array von 0 1“ und platziere ihn 2-mal direkt unterhalb von „beim Start“.

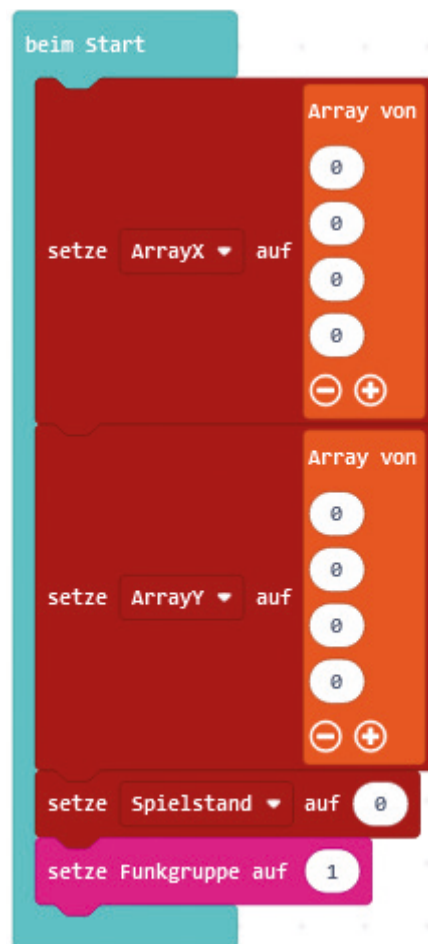
Ändere die 1 in beiden Anweisungen in 0 sowie „list“ in „ArrayX“ und „ArrayY“. Klicke anschließend in beiden „setze list auf Array von 0 1“-Befehlen 2-mal auf das „+“.

Es erscheinen 2 weitere weiße Felder. Nun hast du das Feld gemacht, in dem die X- und Y-Werte der aufleuchtenden LEDs gespeichert werden. Erstelle nun eine neue Variable „Spielstand“.

Setze den Spielstand in „setze Spielstand auf 0“ auf 0.

Dies bedeutet, dass das Spiel noch nicht losgegangen ist.

Zum Schluss musst du Calliope mitteilen, auf welcher Funkfrequenz er sich mit den anderen Calliope unterhalten soll. Dazu benötigst du den Befehl „setze Funkgruppe auf 1“ aus der Gruppe „Funk“.



Wer fängt alle leuchtenden LEDs? (3/9)

Erstelle nun den Code, den du rechts siehst.

Und das passiert im Programm:

- Das Spiel startet, wenn Knopf A gedrückt wird. Dann wird die Variable „Spielstart“ auf 1 gesetzt und dein Calliope sendet den anderen Calliopes eine Zahl, damit sie wissen, dass es losgeht.
- Die Variable „Sieger“ wird auf 3 gesetzt, was in diesem Fall bedeuten soll, dass noch kein Sieger feststeht. (die Variable musst du noch erstellen)
- Die Variable „gefangenePunkte“ soll zählen, wie viele leuchtende starre LEDs du gefangen hast (Variable musst du erstellen). Diese Variable setzt du auf 0 (noch keine LEDs gefangen).

Das blaue Licht der LEDs soll anzeigen, dass das Spiel losgeht.

```

wenn Knopf A geklickt
  wenn Spielstart = 0 dann
    setze Spielstart auf 1
    sende Zahl 0 über Funk
    setze Sieger auf 3
    setze gefangenePunkte auf 0
    setze RGB-LED-Farbe auf [blau]
  
```

Dein Calliope soll 4-mal hintereinander zufällig eine Zahl in X- und Y-Richtung auswählen und die LED dort anschalten. Dafür brauchst du die „für Index von 0 bis 4 mache“-Schleife. Du findest sie unter „Schleifen“. Platziere sie in der „wenn Knopf A geklickt“-Anweisung in den Block „wenn wahr dann“.

Füge in die „für Index von 0 bis 4 mache“-Schleife die Blöcke „setze ZufallspunktX auf“ und „setze ZufallspunktY auf“ ein.

Suche den Block „list füge ein bei 0 Wert“. Er steht unter „Arrays“. Füge ihn unter „setze ZufallspunktX auf“ ein. Ändere „list“ in „ArrayX“ und die 0 in „Index“ (unter „Variablen“).

Füge hinter „Wert“ die Variable „ZufallspunktX“ ein.

Füge einen weiteren Block „list füge ein bei 0 Wert“ unter „setze ZufallspunktY auf“ ein. Ändere „list“ in „ArrayY“ und die 0 in „Index“.

Füge hinter „Wert“ die Variable „ZufallspunktY“ ein. Nun wird der Wert an verschiedenen Stellen in dem Array gespeichert. Dies hängt davon ab, wie oft diese Schleife durchlaufen wurde.

```

setze Sieger auf 3
setze gefangenePunkte auf 0
setze RGB-LED-Farbe auf [blau]
für Index von 0 bis 3 mache
  setze ZufallspunktX auf 0
  ArrayX füge ein bei Index Wert ZufallspunktX
  setze ZufallspunktY auf 0
  ArrayY füge ein bei Index Wert ZufallspunktY
  
```

Hinweis

Sobald du „für Index 0 bis 4 mache“-Schleife in den Code-Bereich ziehst, erstellt der Editor automatisch eine Variable mit dem Namen „Index“. Du brauchst keine neue Variable zu erstellen.

Wer fängt alle leuchtenden LEDs? (4/9)

Lege nun insgesamt 4 Variablen mit dem Namen „ZufallspunktX“, ZufallspunktY“, „Spielstart“ und „gefangene Punkte“ an.

Für die Variablen „ZufallspunktX“ (das ist eine LED-Lampe in einer Reihe) und „ZufallspunktY“ (das ist eine LED-Lampe in einer Spalte) soll Calliope zufällig Zahlen zwischen 0 und 4 auswählen (so viele LED-Lampen befinden sich bei deinem Calliope in einer Reihe/Spalte).

Hierfür brauchst du 2-mal den Block „wähle eine zufällige Zahl von 0 bis 10“. Er steht in der Gruppe „Mathematik“. Ändere die 10 im zweiten weißen Feld in 4 ab. Füge die Blöcke in das weiße Feld mit der 0 bei den Blöcken „setze ZufallspunktX auf“ und „setze ZufallspunktY auf“ ein.

Nun hast du 4-mal hintereinander zufällig eine Zahl in X- und Y-Richtung ausgewählt.

Nun muss Calliope an diesen 4 Zahlen in X- und Y-Richtung die LED anschalten. Dafür benötigst du den Befehl „Zeichne x 0 y 0“. Diesen findest du in der Gruppe „LED“.

Füge in die beiden weißen Felder mit der 0 2-mal den Block „list füge ein bei 0 Wert“ ein. Er steht unter „Arrays“. Platziere in beiden Anweisungen hinter „Wert“ die Variable „Index“ ein. Die „list“ änderst du im ersten „list füge ein bei 0 Wert“-Block in „ArrayX“ und im zweiten „list füge ein bei 0 Wert“ in „ArrayY“.

Erstelle zwei neue Variablen mit dem Namen „YRichtung“ und „XRichtung“. Diese geben die Position des beweglichen Punkts an.

Füge die Anweisungen „setze YRichtung auf 0“ und „setze XRichtung auf 0“ in dem Block „wenn Knopf A geklickt“ am Schluss ein.

Es fehlen noch die beweglichen Punkte und die Anzeige, ob du gewonnen hast.

Füge in die „dauerhaft“-Schleife den Block „während falsch mache“ ein und ergänze den Code, so wie du ihn rechts siehst.

Spielstart = 1 bedeutet, dass das Spiel läuft. In dem Fall macht Calliope bei der LED das Licht aus – und zwar von dem Punkt, der sich bewegen soll. Das macht er mit dem Befehl: „lösche x XRichtung y YRichtung“.

Anschließend schaut er, wie stark du ihn nach rechts, links, vorn und hinten gekippt hast (Bewegungssensor!) und lässt dort mit dem Befehl „Zeichne x XRichtung y YRichtung“ eine LED aufleuchten.

Wird das häufig hintereinander wiederholt, sieht das so aus, als würde sich der Punkt bewegen. Den „runden 0“ und „verteile 0 von niedrig 0 von hoch 1023 zu niedrig 0 zu hoch 4“-Code findest du unter „Mathematik“.

Wer fängt alle leuchtenden LEDs? (5/9)

Für die Anzeige, ob du gewonnen hast benötigst du den „wenn wahr dann“-Befehl.

Platziere diesen in der „dauerhaft“-Schleife.

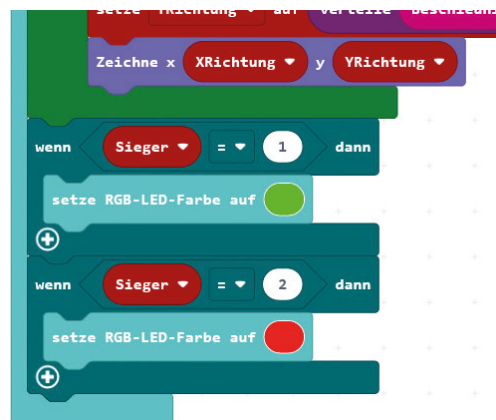
Klicke zweimal auf das „+“ unten links.

Ist in der Variablen „Sieger“ die Zahl „1“ gespeichert, hast du gewonnen und die LED leuchtet grün und Calliope sendet die Zahl „3“ an die anderen Calliopes (das bedeutet, dass du gewonnen hast). Ist die „2“ gespeichert, hast du verloren und die LED leuchtet rot. Ansonsten (zum Beispiel bei Sieger = 3) verändert sich nichts.

Zur Erklärung: Du hattest in den vorherigen Codes festgelegt, dass

- 1 bedeuten soll, dass du gewonnen hast,
- 2 bedeuten soll, dass du verloren hast,
- 3 bedeuten soll, dass noch nicht feststeht, ob du gewonnen hast oder nicht.

Natürlich kannst du den Zahlen auch eine andere Bedeutung zuordnen.



Nun muss du deinem Calliope sagen, was er tun soll, wenn der bewegliche Punkt einen von den 4 Punkten fängt. Die X- und Y-Werte, an denen sich die starren Dots aufhalten, sind in dem Array gespeichert. Daher benötigst du die „für Index von 0 bis 4 mache“-Schleife.

Ändere die 4 in 3.

Erstelle den restlichen Code, wie du ihn unten siehst.

Platziere in diesen Block die Anweisungen „setze ZufallspunktX auf“ und „setze ZufallspunktY auf“.

Die „für Index von 0 bis 4 mache“-Anweisung wird 4-mal durchlaufen, jedes Mal erhöht sich der Index um 1 – also von 0 auf 1, dann auf 2 und auf 3. Da sich der Wert des Index erhöht, schaut Calliope auch jedes Mal an eine andere Position in dem Array. Die Zahl, die sich an der 0., 1., 2., oder 3. Stelle des Arrays befindet, wird dann in der Variablen „Zufallspunkt“ gespeichert.

Dann vergleicht Calliope, ob der „Zufallspunkt“ mit dem aktuellen Aufenthalt des beweglichen Punkts übereinstimmt. Wenn ja, ändert er die Zahl der gefangenen Punkte um 1 und an der Stelle des Arrays, an der er die Zahl gefunden hat, wird sie „herausgelöscht“, also entfernt.

Füge dazu an der Position eine Zahl ein, die kleiner ist als 0, zum Beispiel -20.

Aus der Gruppe „Arrays“ benötigst du die Blöcke „list rufe Wert ab bei 0“ und „list wert festlegen bei 0 auf“.

Wer fängt alle leuchtenden LEDs? (6/9)

Füge in den Block „list wert festlegen bei 0 auf“ den Block „list finde Position von“ ein.

Damit du in dem Feld nach „auf“ von „list Wert festlegen bei 0 auf“ einen Wert eintragen kannst, brauchst du ein weißes Feld „0“. Dieses findest du unter „Mathematik“.

Füge den Befehl „pausiere (ms) 100“ nach der „für Index von 0 bis 4 mache“-Anweisung ein.

Dann macht Calliope, wenn er „für Index von 0 bis 4 mache“ 1-mal durchlaufen hat, eine kurze Pause und schaut, ob ein anderer Calliope etwas gefunkt hat. Erst dann durchläuft er die „für Index von 0 bis 4 mache“-Anweisung erneut.

Jedes Mal, wenn 1 Punkt gefangen wurde, erhöht sich die Variable „gefangenePunkte“ um 1. Wurden alle 4 Punkte gefangen, beträgt die Variable 4. Damit ist sie – wie du sicher aus Mathe schon weißt – größer als 3.

Zur Info: Du kannst auch „gefangenePunkte = 4“ einfügen. Programmierer verwenden jedoch lieber das >-Zeichen als das ==-Zeichen. Denn sollte mal etwas bei einem Programm schiefgehen und bei der Rechnung anstatt 4,5 oder 4,001 herauskommen, dann entsteht bei „gefangenePunkte = 4“ eine Endlosschleife.

Um dies zu vermeiden, nimmt man lieber das >-Zeichen.

Ist die Variable größer 3, soll Calliope einen Ton spielen, der den anderen Mitspielern anzeigt, dass das Spiel beendet ist.

Er sendet zudem den anderen Calliopes die Zahl 3, die diesen anzeigt, dass das Spiel beendet ist.

Die Variable „Sieger“ wird auf 1 gesetzt, d. h., du hast gewonnen.

Der „Spielstart“ wird auf 0 gesetzt, da das Spiel nun vorbei ist.

Wer fängt alle leuchtenden LEDs? (7/9)

```

setze YRichtung auf runden verteile Beschleunigung (mg) y von niedrig -1023 von hoch 1023 zu nie
Zeichne x XRichtung y YRichtung
für Index von 0 bis 3
mache
  setze ZufallspunktX auf ArrayX rufe Wert ab bei Index
  setze ZufallspunktY auf ArrayY rufe Wert ab bei Index
  wenn XRichtung - ZufallspunktX und YRichtung - ZufallspunktY dann
    ändere gefangenePunkte um 1
    ArrayX Wert festlegen bei ArrayX finde Position von XRichtung auf -20
  wenn gefangenePunkte > 3 dann
    spiele Note Mittleres C für 1 Schlag
    setze Sieger auf 1
    setze Spielstart auf 0
  pausiere (ms) 10
  
```

Erstelle nun den Code, den du rechts siehst. Klicke im „wenn wahr dann“-Code auf das kleine + links unten. Es erscheint ein „ansonsten“-Teil.

Empfängt dein Calliope nun die Nummer 0, zeigt das, dass jemand anderes das Spiel gestartet hat. Wurde das Spiel bei deinem Calliope noch nicht gestartet (Spielstart = 0), dann setzt er die Variablen „Spielstart“ auf 1, „Sieger“ auf 3 und „gefangenePunkte“ auf 0 (Erklärung oben). Zudem schaltet die LED auf Blau um.

Ansonsten (wenn er die Nummer „3“ empfängt), wird die Variable „Sieger“ auf 2 gesetzt. Das heißt, dass jemand anderes alle LEDs gefangen und das Spiel als Erstes beendet hat. Du hast verloren.

Die LED-Farbe schaltet auf Rot und das Spiel ist beendet (Spielstart = 0). (Den „setze RGB-LED-Farbe auf“-Befehl benötigst du in dem Code, der mit „beim Start“ beginnt).

```

wenn Zahl empfangen receivedNumber
  wenn receivedNumber = 0 dann
    wenn Spielstart = 0 dann
      setze Spielstart auf 1
      setze Sieger auf 3
      setze gefangenePunkte auf 0
      setze RGB-LED-Farbe auf Blau
    ansonsten
      setze Sieger auf 2
      setze Spielstart auf 0
  
```

Hinweis

„receivedNumber“ musst du nicht neu erstellen. Klicke einfach mit der rechten Maustaste bei „wenn Zahl empfangen receivedNumber“ oben rechts auf „receivedNumber“ und ziehe sie dorthin, wo du sie haben willst.

Wer fängt alle leuchtenden LEDs? (8/9)

Dupliziere nun die „für Index von 0 bis 4 mache“-Schleife in dem „wenn Knopf A geklickt“-Block.

Füge diese unterhalb von „setze RGB-LED-Farbe auf“ in die zweite „wenn wahr dann“-Anweisung ein.

Dies machst du, damit auch der Calliope, der nicht gestartet wurde, vier LEDs zufällig auswählt und speichert.

Füge zudem die Anweisungen „setze YRichtung auf 0“ und „setze XRichtung auf 0“ nach der „für Index von 0 bis 4 mache“-Schleife ein.

```

setze gefangenePunkte auf 0
setze RGB-LED-Farbe auf
für Index von 0 bis 3
  mache
    setze ZufallspunktX auf wähle eine zufällige Zahl von 0 bis 4
    ArrayX füge ein bei Index Wert ZufallspunktX
    setze ZufallspunktY auf wähle eine zufällige Zahl von 0 bis 4
    ArrayY füge ein bei Index Wert ZufallspunktY
Zeichne x ArrayX rufe Wert ab bei Index y ArrayY rufe Wert ab bei Index
setze YRichtung auf 0
setze XRichtung auf 0
ansonsten
  setze Sieger auf 2
  
```

Schließe deinen Calliope über das USB-Kabel an deinen Computer an, speichere das Programm und übertrage es.

Nimm den Calliope vom Computer und verbinde ihn mit dem Batteriehalter.

Starte dein Programm und teste dein Spiel.

Wer fängt alle leuchtenden LEDs? (9/9)

Der komplette Code:

```

    beim Start
    setze list auf [0, 0, 0, 0, 0]
    setze list auf [0, 0, 0, 0, 0]
    setze Spielstand auf 0
    setze Ranggruppe auf 1

    dauerhaft
    wenn Sieger == 1 dann
    sende Zahl 3 über Funk
    setze RGB-LED-Farbe auf [0, 0, 0]
    sonst wenn Sieger == 2 dann
    setze RGB-LED-Farbe auf [0, 0, 0]
    ansonsten
    während
    mache
    lösche x Richtung y Wichtung
    setze Xrichtung auf runden (beschleunigung (mg) x) von niedrig -1023 zu niedrig 0 zu hoch 4
    setze Yrichtung auf runden (beschleunigung (mg) x) von niedrig -1023 zu niedrig 0 zu hoch 4
    zeichne x Richtung y Wichtung
    für Index von 0 bis 3
    mache
    setze ZufallspunktX auf ArrayX rufe Wert ab bei Index
    setze ZufallspunktY auf ArrayY rufe Wert ab bei Index
    wenn Xrichtung == ZufallspunktX und Yrichtung == ZufallspunktX dann
    ändere gefangenePunkte um 1
    ArrayX wert festlegen bei ArrayX finde position von Xrichtung auf -20
    wenn gefangenePunkte > 3 dann
    spiele Note Mittleres-C für 1 Schlag
    setze Sieger auf 1
    setze Spielstart auf 0
    pauseiere (ms) 10
  
```

```

    wenn Knopf A gedrückt
    wenn Spielstart == 0 dann
    setze Spielstart auf 1
    setze Sieger auf 0
    setze gefangenePunkte auf 0
    sende Zahl 0 über Funk
    setze RGB-LED-Farbe auf [0, 0, 0]
    für Index von 0 bis 3
    mache
    setze ZufallspunktX auf wähle eine zufällige Zahl von 0 bis 4
    ArrayX füge ein bei Index wert ZufallspunktX
    setze ZufallspunktY auf wähle eine zufällige Zahl von 0 bis 4
    ArrayY füge ein bei Index wert ZufallspunktY
    zeichne x ArrayX rufe Wert ab bei Index y ArrayY rufe Wert ab bei Index
    setze Wichtung auf 1
    setze Wichtung auf 1
  
```

```

    wenn Zahl empfangen receiveNumber
    wenn receiveNumber == 0 dann
    wenn Spielstart == 0 dann
    setze Spielstart auf 1
    setze Sieger auf 0
    setze gefangenePunkte auf 0
    setze RGB-LED-Farbe auf [0, 0, 0]
    für Index von 0 bis 3
    mache
    setze ZufallspunktX auf wähle eine zufällige Zahl von 0 bis 4
    ArrayX füge ein bei Index wert ZufallspunktX
    setze ZufallspunktY auf wähle eine zufällige Zahl von 0 bis 4
    ArrayY füge ein bei Index wert ZufallspunktY
    zeichne x ArrayX rufe Wert ab bei Index y ArrayY rufe Wert ab bei Index
    setze Wichtung auf 0
    setze Wichtung auf 0
    ansonsten
    setze Sieger auf 2
    setze Spielstart auf 0
  
```