

Weihnachtsprojekt: Verteiltes Glockenspiel mit mehreren Calliope Mini

Originalartikel

Backup

<html> <time datetime=„2018-11-25“>heute</time> Bernd Heisterkamp <figure class=„aufmacherbild“></figure><p class=„lead_text“>Ein oder besser mehrere Calliope mini sorgen mit Aluminiumrohren und einer ausfeilten Programmierung für weihnachtliche Musik im Wohnzimmer.</p><p>Zu Weihnachten geht es festliche Musik wie Lebkuchen, Mandarinenteller und Geschenke. Wer sich lieber nicht mit der Blockflöte vor den Tannenbaum stellen, aber gerne etwas selbst machen möchte, sollte unser Glockenspiel mit dem Mikrocontroller Calliope mini bauen.</p> <figure class=„rteinlinebild akwa-inline-img rtepos_right col-lg-6 col-md-6 col-sm-6 col-xs-12 akwa-inline-right“ is=„a-grossbild“><a class=„grossbild cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„/make/imgs/76/2/5/3/9/8/4/1/01-Calliope_Glockenspiel_Uebersicht-d899a1a7ea0c1313.JPG“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=16>“></figure><p>Die Christbaumanhänger machen Musik, indem sie Klappe gegen Aluminiumrohre schlagen. Jeder Anhänger hat dabei zwei unterschiedlich lange Rohre und kann somit zwei unterschiedliche Töne erzeugen. Durch Variation der Rohrängen der Anhänger kann die ganze Tonleiter abgedeckt werden. Ein beliebiger Calliope versendet die Tonfolge im Takt der Musik per Bluetooth, die alle anderen empfangen. Nur die Calliopes, die den aktuellen Ton als ihren eigenen erkennen, spielen diesen dann darüber den Klappschlag ab. Je mehr Calliopes beteiligt sind, desto schöner kann die Melodie über den ganzen Weihnachtsbaum verteilt werden.</p> <p>Dieses Projekt richtet sich an Lehrkräfte oder Eltern, die einer Gruppe von Kindern das Programmieren näher bringen möchten. Neben dem Programmieren von Mikrocontroller und Servo gilt es, mit der Laubsäge zu arbeiten und etwas darüber Musik zu lernen. Beginnend mit einem einfachen Programm werden zunehmend komplexere Funktionen verwendet, von Schleifen darüber Arrays bis hin zu Funktionen und Text-String-Analyse, was etwas Erfahrung mit der PXT-Blocksprache bedarf.</p> <div class=„rtetextbox akwa-inline-textbox col-lg-12 col-md-12 col-sm-12 col-xs-12“><h4>Kurzinfo</h4> <h4>Material pro Anhänger</h4> ein Calliope miniBatteriefach mit 2 AA-Batterien
 Aluminiumrohr (14,8 mm, 1 mm Wandstärke), Länge je nach gewünschten Tonhöhen
 Sperrholz (3 mm dick) ein Holzstab (90 mm lang)
 eine M5-Schraube (60 mm lang, flacher Kopf) zwei Schrauben mit Unterlegscheiben und Muttern
 Schrumpfschlauch (mind. 30 mm)
 etwas Schnur, möglichst transparent <p>Checkliste:</p>Zeitaufwand: 1 bis 2 Stunden für den Zusammenbau

Kosten: 50 Euro
 Holzbearbeitung: einfache Sägearbeiten
 Programmieren: Calliope-MakeCode-Editor
 </div> <noscript readability=„1“> <p>Bei Problemen mit der Wiedergabe des Videos aktivieren Sie bitte JavaScript</p> </noscript> <figure class=„video video-fullwidth akwa-inline-video player“><div class=„kultura-video-container kultura-player-shell js-player-dummy-play-button“></div> </figure> <p>Die Anhänger werden aus 3 mm dicken Sperrholz geschnitten. Wer Zugang zu einem Lasercutter hat, kann die Sterne mit ausschneiden, bei der Arbeit mit einer Laubsäge sollten sie aufgemalt werden. Wichtig ist, die Aussparungen für den Servo und die Befestigungsschrauben des Calliope genau zu sägen, bzw. zu bohren.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12“ is=„a-grossbild“><a class=„grossbild cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„[make/imgs/76/2/5/3/9/8/4/1/02-Calliope_Glockenspiel_Material-3e730efd5cac7fad.JPG](https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=17)“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=17>“> <figcaption class=„rteinlinebild_source akwa-caption“>Alle benötigten Teile</figcaption></figure> <p>In die Bohrung über der langen Aussparung des Anhängers wird quer ein Holzstab von etwa 90 mm Länge eingeklebt, an diesem Holzstab werden die Klangkörbe befestigt. Einkerbungen oder kleine Bohrungen am Ende des Holzstabs dienen der Fixierung der Befestigungsschnur.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12“ is=„a-grossbild“><a class=„grossbild cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„[make/imgs/76/2/5/3/9/8/4/1/03-Calliope_Glockenspiel_Aufhaengung-feb34397e152f987.jpeg](https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=17)“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=17>“> <figcaption class=„rteinlinebild_source akwa-caption“>Die Aufhängung der Metallrohre</figcaption></figure> <p>Die Klangkörper werden von einem Aluminiumrohr abgesägt, die Länge und Dicke des Rohrs beeinflusst die Tonhöhe, je kürzer das Rohr, desto höher der Ton. In der Tabelle seht ihr, welche Längen von einem 14,8 mm dicken Rohr mit 1 mm Wandstärke für die Grundtöne der Tonleiter abgeschnitten werden müssen. Wer ein anderes Metallrohr nimmt, kann die richtigen Längen ausrechnen.</p> <p><table class=„rtable table akwa-inline-table“><tbody><tr><td class=„c2“>Ton</td> <td class=„c2“>Frequenz</td> <td class=„c2“>Länge</td> </tr></tbody></table></p>

class=„c2“>659,255</td> <td class=„c2“>374</td> </tr><tr><td class=„c2“>d</td> <td class=„c2“>587,33</td> <td class=„c2“>397</td> </tr><tr><td class=„c2“>c</td> <td class=„c2“>523,251</td> <td class=„c2“>420</td> </tr><tr><td class=„c2“>h</td> <td class=„c2“>493,883</td> <td class=„c2“>433</td> </tr><tr><td class=„c2“>a</td> <td class=„c2“>440</td> <td class=„c2“>458</td> </tr><tr><td class=„c2“>g</td> <td class=„c2“>391,995</td> <td class=„c2“>486</td> </tr><tr><td class=„c2“>f</td> <td class=„c2“>349,228</td> <td class=„c2“>514</td> </tr><tr><td class=„c2“>e</td> <td class=„c2“>329,625</td> <td class=„c2“>529</td> </tr><tr><td class=„c2“>d</td> <td class=„c2“>293,665</td> <td class=„c2“>561</td> </tr><tr><td class=„c2“>c</td> <td class=„c2“>261,626</td> <td class=„c2“>594</td> </tr></tbody></table></p> <figure class=„rteinlinebild akwa-inline-img rtepos_right col-lg-6 col-md-6 col-sm-6 col-xs-12 akwa-inline-right“ is=„a-grossbild“><a class=„grossbild cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„/make/imgs/76/2/5/3/9/8/4/1/18_Bild_FFT-a950ccaa362464b2.jpg“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=34>“></figure><p>Dazu nutze ich die Gleichung: $L = \frac{A}{f}$. Nach einem Instructables-Projekt ist die Rohrlänge und die gewünschte Frequenz. A ist eine Konstante, die nur sehr schwierig zu berechnen ist, aber leicht gemessen werden kann. Nach Umformung kommt man auf die Gleichung $A = f L^2$. Kennt man die Schwingungsfrequenz einer bestimmten Rohrlänge, kann man so A bestimmen. Zum Glück hat heute jeder einen Frequenzanalysator in der Hosentasche, ladet dazu eine App zur Fast Fourier Transformation (FFT) auf euer Handy (z.B. <a href=„<https://itunes.apple.com/de/app/fftwave/id1080227446?mt=8>“ rel=„external“ target=„blank“>FFTWave für iOS).</p> <p>Geht nun wie folgt vor:</p> <ol class=„rtelist rtelist-ordered“>Sgt ein Rohr von 50 cm Länge ab, hängt es einer dünnen Perlonschnur auf und bringt es mit einem Stab zum Erklingen. Sobald der Klang etwas abgeklungen ist, startet die App und speichert das gemessene Spektrum (siehe Bild). Die niedrigste Frequenz sollte der Grundschwingung entsprechen, im Beispiel 495,3Hz. Berechnet $A = f L$, hier $A = 495,3 \text{ Hz} \cdot (50 \text{ cm}) = 1238,25 \text{ cm}$ und berechnet die Rohrlänge gemäß $L = \frac{A}{f}$, also $L = \frac{495,3 \text{ Hz} \cdot 50 \text{ cm}}{440 \text{ Hz}} = 53 \text{ cm}$ <p>Als Kloppe nehmen ihr eine M5-Schraube. Sie wird über einen Schrumpfschlauch mit dem Ruderhorn verbunden. Dazu gibt ihr einen Tropfen Alleskleber auf das Horn, stellt einen Schlauch darüber und fügt auf der offenen Seite die Schraube hinein. Die Schraube darf das Ruderhorn nicht berühren, sondern es müssen etwa 5 mm Abstand bleiben, dadurch kann die Schraube etwas über die Endposition des Servos hinausschwingen und das Rohr ganz kurz antippen. Den Schrumpfschlauch mit Haar- oder Heißluftfalten bearbeiten, bis er das Gewinde der Schraube und das Ruderhorn so stark einquetscht, dass diese nicht mehr rausrutschen kann. Der Schrumpfschlauch hat sich als optimal herausgestellt, da er genug Stabilität bietet, die Schraube über den Servo zu bewegen, aber gleichzeitig ein Schwingen der Schraube ermöglicht.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12“ is=„a-grossbild“>

href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html>“><img alt=„Metallklöppel aus Schraube, Schrumpfschlauch und Rotorhorn“

src=„https://heise.cloudimg.io/width/2000/q50.png-lossy-50.webp-lossy-50.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/04-Calliope_Glockenspiel_Kloeppel-c1c3414d5c30a5d2.jpeg“

srcset=„https://heise.cloudimg.io/width/4000/q30.png-lossy-30.webp-lossy-30.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/04-Calliope_Glockenspiel_Kloeppel-c1c3414d5c30a5d2.jpeg 2x“

class=„c1“> <figcaption class=„rteinlinebild_source akwa-caption“>Der fertige Klöppel aus Schraube, Schrumpfschlauch und Rotorhorn</figcaption></figure><p>Der Servo wird mit einem Kabelbinder auf dem Holzanhänger befestigt. Wenn ihr nicht löten möchtet, kauft ihr am Besten einen Servo mit Grove-Stecker, der direkt an den Calliope gesteckt wird.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12“ is=„a-grossbild“><a class=„grossbild cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„/make/imgs/76/2/5/3/9/8/4/1/05-Calliope_Glockenspiel_Servo-01e9ee5e60d32731.JPG“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html>“>

src=„https://heise.cloudimg.io/width/5472/q50.png-lossy-50.webp-lossy-50.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/05-Calliope_Glockenspiel_Servo-01e9ee5e60d32731.JPG“

srcset=„https://heise.cloudimg.io/width/10944/q30.png-lossy-30.webp-lossy-30.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/05-Calliope_Glockenspiel_Servo-01e9ee5e60d32731.JPG 2x“

class=„c1“> <figcaption class=„rteinlinebild_source akwa-caption“>Servo und Klöppel</figcaption></figure><p>Ansonsten müsst ihr den Stecker separat kaufen und anlöten. Dabei müsst ihr auf die richtige Verkabelung achten. Das Servokabel hat drei Adern: braun oder schwarz, rot und gelb oder weiß. In der Grafik seht ihr, wie der Servo an die Grove -Steckerbuchse A1 (sie befindet sich über dem Button B) angeschlossen werden muss. Anschließnd die Lötstellen mit Schrumpfschlauch isolieren, ansonsten gibt es einen Kurzschluss.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12 c3“>

src=„https://heise.cloudimg.io/width/613/q50.png-lossy-50.webp-lossy-50.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/06_Calliope_Glockenspiel_ServoAnschluss-dafb8c5bdcf4e2a8.jpg“

srcset=„https://heise.cloudimg.io/width/1226/q30.png-lossy-30.webp-lossy-30.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/06_Calliope_Glockenspiel_ServoAnschluss-dafb8c5bdcf4e2a8.jpg 2x“

class=„c1“><figcaption class=„rteinlinebild_source akwa-caption“>Servo-Anschlüsse für den Grove-Stecker</figcaption></figure> <p>Zuerst soll der Calliope so programmiert werden, dass sich der Klöppel beim Drücken des Knopfes A kurz nach vorne und direkt wieder in die Mitte bewegt. Beim Drücken von B er nach hinten und ebenfalls wieder in die Mittelposition gefahren. So könnt ihr probieren, ob der Klöppel die Rohre sauber anschlägt.</p> <figure class=„rteinlinebild akwa-inline-img rtepos_left col-lg-6 col-md-6 col-sm-6 col-xs-12 akwa-inline-left c4“>

src=„https://heise.cloudimg.io/width/306/q50.png-lossy-50.webp-lossy-50.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/07_Bild_Konzert-09aa820448e9885f.PNG“

srcset=„https://heise.cloudimg.io/width/612/q30.png-lossy-30.webp-lossy-30.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/07_Bild_Konzert-09aa820448e9885f.PNG 2x“ class=„c1“><figcaption class=„rteinlinebild_source akwa-caption“>Konzert 1</figcaption></figure><p>Den Befehl <code>Wenn Knopf A gedrückt</code> findet ihr im Menü „Eingabe“. Alle Aktionen, die in der Klammer dieses Befehls stehen, werden beim Drücken des Knopfes ein Mal ausgeführt. Als Erstes soll der Servo in eine neue Position fahren, dazu braucht ihr den Befehl <code>Schreibe Servo an Pin … auf …</code> aus dem Menü „Fortgeschritten -> Pins“. Voreingestellt ist <code>P1</code>, stattdessen müsst ihr <code>C17</code>

auswählen, da ihr an diesem Pin das gelbe oder weiße Kabel des Servos befestigt habt. Als Zielwert nehmt ihr zunächst <code>75</code>, was eine Änderung von 15 Grad aus der Mittelposition <code>90</code> bedeutet. Mit dem Befehl <code>Pausiere (ms) 100</code> gibt ihr dem Servo Zeit, um die Position anzufahren, bevor ihr ihn mit <code>Schreibe Servo an Pin C17 auf 90</code> wieder in die Mitte bewegt. Für <code>Wenn Knopf B gedrückt</code> werden die Programmierbefehle wiederholt; allerdings soll der Servo in die andere Richtung fahren, daher wird als Zielwert <code>112</code> vorgegeben. </p> <p>Nun kann ihr das Programm auf den Calliope übertragen und die Batterieversorgung anschließen. Der USB-Anschluss alleine reicht nicht für die Stromversorgung des Servos. Variiert zur Überraschung die Werte, die ihr dem Servo vorgebt (im Beispiel <code>75</code> und <code>112</code>) bis der Klang abgespielt bis kurz vor die Klangrohre fällt, diese aber nicht berichtet. Sonst wird der Ton nach dem Anschlagen direkt wieder gedämpft. Die Pausendauer muss so angepasst werden, dass der Servo die maximale Position auch erreicht, bevor er den Befehl zum Zurückfahren erhält. </p> <p>Habt ihr alles richtig eingestellt, kann mit dem Calliope durch Drücken der Knöpfe A und B zwei unterschiedliche Töne erzeugen. In Schulklassen eine gute Gelegenheit, um gemeinsam eine kleine Melodie auszuprobieren, denn jeder Calliope sollte zwei andere Töne erzeugen. </p> <p>Es wäre allerdings unpraktisch, wenn die ganze Gruppe am Weihnachtsbaum stehen und die Knöpfe im richtigen Takt drücken müssen. Daher ist jetzt das Ziel, die Noten über die Bluetooth-Schnittstelle kabellos zu übertragen. Die Melodie wird dazu in einer Liste gespeichert, in der Programmierung ist dies ein Array. Ihr findet die Funktion im Fortgeschritten-Menü des Calliope Editors unter „Arrays“. </p> <p>Erstellt zunächst eine Variable mit dem Namen <code>MeineMelodie</code> (Menü „Variablen -> Neue Variable anlegen“). Dieser weist ihr mit den Befehlen <code>ändern Wert auf MeineMelodie</code> und <code>Erstelle Array mit einem noch leeren Array zu. Über das blaue Setup-Symbol im Befehl <code>Erstelle Array mit ...</code> kann die Anzahl der Listenelemente geändert werden. Zieht einfach so viele Elemente in das Array wie euer Lied Noten hat. In die Elemente schreibt ihr eure Melodie von oben nach unten. Die Ziffern neben den Noten beschreiben die Oktave. Im Beispiel sind alle aus der gleichen Oktave. Wenn ihr ein Lied über mehrere Oktaven habt, müsst ihr die Ziffern entsprechend anpassen (C1, C2, ... usw.). </p> <caption>Konzert 2: Melodie senden</caption> <p>Mit einer <code>Für Index von 0 bis 4</code> Programmierschleife (siehe Menü „Schleifen“) soll nacheinander jedes Element des Arrays ausgelesen werden. Die Voreinstellung <code>4</code> wird mit der Variablen <code>Anzahl</code> ersetzt, die automatisch die Anzahl der Listenelemente speichert. Da der Index bei <code>0</code> und nicht bei <code>1</code> beginnt, muss für die Anzahl der Schleifendurchläufe noch 1 subtrahiert werden. Das geht mit dem <code>0 < 4</code> Befehl im Menü „Mathematik“. Die erste Null wird mit der Variablen <code>Anzahl</code> und die zweite Null durch eine <code>1</code> ersetzt. </p> <caption>Calliope Mini</caption>

srcset=„https://heise.cloudimg.io/width/420/q30.png-lossy-30.webp-lossy-30.foil1/_www-heise-de_im_gs/71/2/2/5/4/4/7/7/Calliope_mini-9a308b345966bdc2.jpeg“ 2x“ class=„c1“></figure> <p>Der Mikrocontroller Calliope mini wurde in Deutschland entwickelt und soll bereits in der Grundschule Programmierkenntnisse vermitteln.</p> <a href=„<https://www.heise.de/select/make/2017/2/1492870483451450>“ rel=„external“>Calliope mini in Make 2/17 <a data-content-type=„content“ href=„<https://m.heise.de/make/artikel/Calliope-mini-3676938.html>“>Beispielprogramme im Video <a data-content-type=„content“ href=„<https://m.heise.de/meldung/Calliope-mini-Mikrocontroller-fuer-die-Grundschule-3361271.html>“>Betaversion im Praxistest <a data-content-type=„stream“ href=„<https://m.heise.de/thema/calliope-mini>“>Themenseite Calliope mini</div> <p>Alle Befehle, die in der gren Klammer der Schleife stehen, werden nun genauso hufig ausgefrt wie die empfangene Melodie Noten hat. Bei jedem Durchlauf wird die Variable

Index

um

1

erhht und es somit mglich, ein weiteres Element aus der Liste auszulesen. Dazu braucht ihr den Befehl

Liste rufe den Wert ab bei 0

aus dem Men „Arrays“. Anstelle des Wertes

Liste

wlt ihr die Variable

MeineMelodie

und statt der Null setzt ihr die Variable

Index

ein. Mit

–ndere AktuelleNote auf …

wird der zuvor bezogene Listenwert der Variable

AktuelleNote

zugewiesen.</p> <p>Wlt den Befehl

Sende Zeichenfolge

im Menü;

Funk

und wählt die Variable

AktuelleNote

um genau diesen Wert per Bluetooth in die Welt zu senden. Das Senden der nächsten Note soll um die Taktzeit verzögert werden. Hierfür nutzt ihr den Befehl

pausiere (ms) Takt

und setzt als Wert

500

ein. Dies sorgt für eine Unterbrechung von einer halben Sekunde bevor die nächste Note aus dem Array versendet wird. Damit später klar, ist wer zu eurer Gruppe gehört und somit die Noten empfangen soll, muss in der

Beim Start

Anweisung eine Gruppe mit dem Befehl

Setze Gruppe 0

definiert werden (Menü „Funk“ „mehr“).</p> <p>Die unterschiedlichen Calliopes sollen die versendeten Noten mithören, wenn ob sie selbst die empfangenen Noten spielen können und dies gegebenenfalls auch tun. Hierzu müsst ihr zunächst definieren, welche Noten euer Calliope spielen kann. Definiert zwei Variablen

MeineNote1

und

MeineNote2

und weist ihnen mit der Anweisung

Ändere MeineNote1 auf …

(Menü „Variablen“) sowohl für Note 1 als auch für Note 2 die Tonhöhen zu, die ihr mit den Klangrohren erzeugen können. Damit ihr später seht, auf welche Noten euer Calliope eingestellt ist, solltet ihr sie im Startmenü mit dem Befehl

Zeige Zeichenfolge

kurz anzeigen.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12 c6“><img alt=„Screenshot Programm 2 "Empfangen"“

`src=„https://heise.cloudimg.io/width/674/q50.png-lossy-50.webp-lossy-50.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/09_Bild_KonzertNo2_empfangen-6f9c25bef126683a.PNG“`
`srcset=„https://heise.cloudimg.io/width/1348/q30.png-lossy-30.webp-lossy-30.foil1/_www-heise-de_/make/imgs/76/2/5/3/9/8/4/1/09_Bild_KonzertNo2_empfangen-6f9c25bef126683a.PNG_2x“`
`class=„c1“><figcaption class=„rteinlinebild_source akwa-caption“>Konzert 2: Melodie empfangen</figcaption></figure><p>Der Befehl`

Wenn Datenpaket empfangen receivedString

aus dem Menü „Funk“ speichert die empfangene Zeichenfolge in der Variablen

receivedString

. Benennt diese Variable in

AkutelleNote

durch klicken auf

receivedString

um, das macht den Programmtext leichter verständlich.</p> <p>Nun braucht ihr die Funktion

Wenn wahr dann …

Funktion (Menü „Logik“) um zu überprüfen, ob es sich bei der empfangenen Note um eine handelt, die euer Calliope spielen kann. Ebenfalls im Menü „Logik“ befindet sich der Operator

0 = 0

. Ersetzt die erste Null mit

AkutelleNote

und die zweite Null mit

MeineNote1

. Es wird also geprüft, ob die empfangene Note der ersten Note eures Calliopes entspricht. Die Befehle in der grünen Klammer entsprechen nun denen aus Konzert Nummer 1, wenn der Knopf A gedrückt wurde. Den

Wenn wahr dann …

Befehl muss ihr noch für die zweite Note wiederholen und anpassen. Jetzt kann ihr die beiden Programmteile Senden und Empfangen in einem Programm kombinieren. Somit kann jeder Calliope sowohl als Sender als auch als Empfänger eingesetzt werden.</p> <p>Noch kann der Calliope, der die Noten versendet, seine eigenen Klangröhren allerdings nicht

ert-önen lassen sollte. Die Erweiterung ist sehr einfach. Nachdem der Calliope eine Note versendet hat, soll er einfach pr-üfen, ob er selbst die Note spielen kann. Dazu wird der

```
wenn (AktuelleNote = MeineNote1) dann &#8230;
```

Befehl aus dem vorgehenden Beispiel wiederholt und hinter den Befehl

```
Sende Zeichenfolge AktuelleNote
```

gestellt und zwar einmal f-ür

```
MeineNote1
```

und nochmal f-ür

```
MeineNote2
```

. Jetzt kann auch der Dirigent mit musizieren.

</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12“ is=„a-grossbild“><a class=„grossbild_cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„/make/imgs/76/2/5/3/9/8/4/1/11_Bild_KonzertNo3a-f96211cf15baa15d.PNG“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=27>“>

<figcaption class=„rteinlinebild_source akwa-caption“>Konzert 3: Melodie senden und empfangen, mit Funktionen</figcaption></figure><p>Allerdings wiederholen Programmierer nicht gerne Programmcode. Kleine Änderungen m-üssen ansonsten an mehreren Programmstellen eingearbeitet werden, was m-ühsam und fehlertr-ächtig ist. Abhilfe schafft die Verwendung von Funktionen (Men-ü „Fortgeschritten“ „Funktionen“). Der Programmteil, der sich wiederholt, wird in die blaue Klammer der Funktion verschoben und dieser ein Name gegeben. Am Besten einer, der auf die Aufgabe der Funktion schließen l-ässt. In diesem Beispiel bietet es sich an, die Funktionen <code>SpieleNote1</code> und <code>SpieleNote2</code> zu definieren. Sie enthalten die Befehle f-ür die Servosteuerung.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12“ is=„a-grossbild“><a class=„grossbild_cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„/make/imgs/76/2/5/3/9/8/4/1/12_Bild_KonzertNo3b-d6ec2de178289dae.PNG“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=28>“></figure> <p>Sollen nicht nur ganze Noten, sondern auch Halb-, Viertel- und Achtelnoten in der Melodie verwendet werden, muss die Notenl-änge zusammen mit der Tonh-öhe im Array gespeichert werden. Dazu werden vor der Tonh-öhe die Werte <code>g</code> (ganze), <code>h</code> (halbe), <code>v</code> (viertel) oder <code>a</code> (achtel) erg-änzt. Eine Achtelnote D der zweiten Oktave wird somit zu

<code>aD2</code>. Der erste Buchstabe der Note bestimmt die Pausenzeit bis die Note gespielt wird.</p> <p>Definiert drei neue Variablen: <code>Notenlaenge</code>, <code>Ton</code> und <code>Takt</code>. Der Variablen <code>Takt</code> weist ihr in der Startklammer einen fixen Wert zu, beispielsweise 1000 ms. Der Variablen <code>Notenlaenge</code> weist ihr mit dem Befehl <code>Extrahiere aus … beginnend mit …</code> (Menü „Fortgeschritten“ „Text“) den ersten Buchstaben des in <code>AktuelleNoteEmpfangen</code> gespeicherten Texts zu. Dazu müsst ihr die Werte von Position <code>0</code> bis <code>1</code> auswählen. Mit <code>Ändere Notenlaenge auf / Extrahiere aus AktuelleNoteEmpfangen von 0 bis 1</code> wird also die Information g,h,v oder a in <code>Notenlaenge</code> gespeichert. Der <code>Ton</code>, beispielsweise D1, wird mit dem gleichen Befehl und den Positionen <code>1</code> bis <code>3</code> gespeichert.</p> <p>Für die Einhaltung der Pause definiert ihr noch die Funktion <code>Taktpause</code>. Nutzt die <code>Wenn … dann … sonst wenn…</code> Funktion (Menü „Logik“) um die Variable <code>Notenlaenge</code> mit g,h,v und a zu vergleichen. Dazu müsst ihr im Setup der Funktion (blaues Icon) weitere <code>else if</code> Optionen in die Funktion ziehen. Den Wert in der Variablen <code>Notenlaenge</code> vergleicht ihr nach nacheinander mit den möglichen Tonlängen. Ist die richtige Tonlänge gefunden, wird der Pausenbefehl ausgeführt. Die Länge der Pause entspricht entweder dem <code>Takt</code> (Länge g), dem <code>Takt ÷ 2</code> (h), dem <code>Takt ÷ 4</code> (v) oder dem Takt geteilt durch 8 (a). Der Teilen-Befehl befindet sich im „Mathematik“ Menü.</p> <caption>Konzert 3 mit alben, Viertel- und Achtel-Noten. Aus Gründen der Übersichtlichkeit wurde dem Calliope-Dirigenten das Instrument weggenommen. Der Calliope, der eine Melodie sendet, spielt selbst keinen Ton. Aber das kann natürlich leicht geändert werden.</caption> <p>Ein Weihnachtsbaumänger sollte auch schön leuchten. Folgende Funktionen sollen zum Schluss noch programmiert werden:</p> <ol class="rteList_rteList-ordered"> Jedes Mal, wenn eine Note empfangen wird, soll die bunte LED des Calliope in einer zufällig ausgewählten Farbe leuchten. Die LED-Matrix soll einen mit zwei Bildern animierten Stern anzeigen. <caption>Konzert 5: RGB-LED

<https://schnipsl.qgelm.de/>

Printed on 2025/08/05 00:51

beleuchten</figcaption></figure><p>Erstellt für den Farbwechsel eine eigene Funktion und nennt sie

ZeigeZufaelligeFarbe

und erstellt eine neue Variable mit dem Namen

Farbe

. Dieser Variablen wird mit Befehl

Wählen eine zufällige Zahl zwischen 0 und 4

(Menü „Mathematik“) ein Zufallswert zugewiesen. Mit

Wenn ; dann ; sonst wenn;

Funktion wird die Variable mit den fünf möglichen Werten verglichen und die LED, je nach Zahlenwert, mit dem Befehl

setze LED-Farbe auf ;

(Menü „Grundlagen“) auf eine andere Farbe eingestellt. Die Funktion wird auch im Programmteil

wenn Knopf A gedrückt

vor der

Taktpause

aufgerufen.</p> <figure class=„rteinlinebild akwa-inline-img rtepos_left col-lg-6 col-md-6 col-sm-6 col-xs-12 akwa-inline-left“ is=„a-grossbild“><a class=„grossbild cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„/make/imgs/76/2/5/3/9/8/4/1/14_Bild_KonzertNo5_Animation-dd12de94731b97cc.PNG“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=30>“> <figcaption class=„rteinlinebild_source akwa-caption“>Konzert 5: LED-Matrix animieren</figcaption></figure><p>Für die Animation auf der LED-Matrix erstellt ihr eine weitere Variable namens

Animation

. Dieser Variable weist ihr den Befehl

erstelle großes Bild

(Men-> „Fortgeschritten“ -> „Bilder“) zu. Zeichnet in den ersten f<nf Spalten den ersten Teil der Sternanimation und in den hinteren f<nf Spalten den zweiten Teil. Aufgerufen wird die Animation über den Befehl

scroll Bild Animation mit Offset … und Intervall … (ms)

. Der Offset bestimmt, um wie viele Spalten das Bild bei der wiederholten Anzeige verschoben wird. Bei

1

würden die Bildanimationen wie ein Lauftext nach links scrollen. Stellt den Wert auf

5

, so sieht man erst das eine, dann das andere Bild. Die Blinkfrequenz wird über den Intervallwert eingestellt. Da diese Animation dauerhaft laufen soll, also auch wenn keine Melodie gespielt wird, wird der Befehl in der Schleife

dauerhaft

eingebettet.</p> <figure class=„rteinlinebild akwa-inline-img akwa-inline-img col-lg-12 col-md-12 col-sm-12 col-xs-12“ is=„a-grossbild“><a class=„grossbild cbox_gallery akwa-inline-imglink“ data-content-type=„image“ data-grossbildsrc=„/make/imgs/76/2/5/3/9/8/4/1/17_microbit-KonzertNo5-ebbec0bd4ffef005.PNG“ href=„<https://m.heise.de/make/artikel/Weihnachtsprojekt-Verteiltes-Glockenspiel-mit-mehreren-Calliope-Mini-4219883.html?view=zoom;zoom=33>“></figure><p>Das war's, euer Weihnachtsbaumhänger ist fertig. Viel Erfolg beim Nachbauen!</p> <noscript readability=„1“> <p>Bei Problemen mit der Wiedergabe des Videos aktivieren Sie bitte JavaScript</p> </noscript> <figure class=„video video-fullwidth akwa-inline-video player“><div class=„kaltura-video-container kaltura-player-shell js-player-dummy-play-button“></div> </figure><ul class=„rte_list rte_list-unordered“>Hier gibt es alle Dateien zum Download, inklusive der Plot-Vorlagen und Hex-Dateien für den Calliope sowie eine Version für den britischen BBC micro:bit. <div class=„btn-toolbar whatsbroadcast-toolbar“> <p>Teilen </p> <p>Abonnieren </p> </div> </html>

From:

<https://schnipsl.qgelm.de/> - **Qgelm**

Permanent link:

https://schnipsl.qgelm.de/doku.php?id=wallabag:weihnachtsprojekt_-verteiltes-glockenspiel-mit-mehreren-calliope-mini

Last update: **2021/12/06 15:24**

