

Zeichencodierung mit dem Calliope

Didaktische Hinweise

Didaktische Einordnung

Das Thema Codierung spielt in der Informatik in vielen Teilgebieten eine Rolle und wird den Schüler*innen in unterschiedlichen Kontexten begegnen. Ziel der hier vorgestellten Unterrichtsreihe ist zunächst nicht das Erlernen einer speziellen Codierung, wie z. B. dem ASCII-Code, sondern ein Grundverständnis für die Notwendigkeit und den Aufbau von Codierungen zu entwickeln. Da Computer Binärcodes verwenden, werden auch hier von den Schüler*innen binäre Codes entwickelt. Durch das Testen und Vergleichen ihrer eigenen Codes, können die Schüler*innen verschiedene Erkenntnisse über den Aufbau von Codierungen erlangen (vgl. Lernziele). Erst wenn die Schüler*innen Einblicke in die Gestaltungsmöglichkeiten von Codierungen bekommen haben, wird auf dieser Grundlage der ASCII-Code und ggf. der Morse-Code betrachtet.

Als Werkzeug wird hier der Calliope verwendet, da er durch die Beschränkung auf zwei Tasten eine binäre Codierung von Zeichen nahelegt. Auf diese Weise kann die Codierung von den Schüler*innen direkt ausprobiert werden und ist damit greifbarer als binäre Codierungen, die innerhalb des Rechners verwendet werden.

Um den Schüler*innen unterschiedlicher Jahrgänge und mit unterschiedlich vielen Vorerfahrungen im Bereich Algorithmik einen Zugang zu dem Thema zu ermöglichen, werden verschiedene Arbeitsblätter mit unterschiedlich großen Programmieranteilen angeboten. Liegen noch keine Programmierkenntnisse vor, kann der Calliope mit einem fertigen Programm auch ganz ohne Implementierung auf Seiten der Schüler*innen eingesetzt werden. Die Codierung der Zeichen wird bei allen Varianten von den Schüler*innen vorgenommen, um verschiedene Zeichen mithilfe der Tasten A und B in den Calliope eingeben zu können. Während in Variante 1 und 2 auch die Decodierung von den Schüler*innen vorgenommen wird, wird diese in Variante 3 algorithmisch umgesetzt.

Auch für die anschließende Auseinandersetzung mit standardisierten Codes am Beispiel des ASCII-Codes kann zwischen zwei Graden der Vertiefung gewählt werden.

Zielgruppe

Die vorliegenden Materialien können in den Jahrgängen 6 bis 10 eingesetzt werden. Je nach Vorkenntnissen der Schüler*innen ist ein Zugang mit einem angemessenen Anteil an Algorithmik auszuwählen.

Voraussetzungen

Es werden zwar binäre Codes entwickelt, diese werden jedoch zunächst nicht als Binärzahlen interpretiert. Das binäre Zahlensystem wird daher nicht als bekannt vorausgesetzt. Soll vertiefend eine Interpretation des binären Codes als binäre Zahl erfolgen, wird das binäre Zahlensystem an entsprechender Stelle eingeführt.

Abhängig von den Vorkenntnissen im Bereich Algorithmik wird das erste Arbeitsblatt in drei verschiedenen Varianten angeboten:

Variante 1: Variante 1 sieht vor, dass auf dem Calliope bereits ein Programm gespeichert ist, das bei Drücken der Taste A eine 0 und bei Drücken der Taste B eine 1 an einen anderen Calliope überträgt. Somit kann der Calliope zum Übertragen von Codierungen aus 0en und 1en verwendet werden, ohne dass die Schüler*innen selbst etwas programmieren müssen.

Variante 2: Variante 2 sieht vor, dass die Schüler*innen selbst ein Programm implementieren, welches bei Drücken der Taste A eine 0 und bei Drücken der Taste B eine 1 an einen anderen Calliope überträgt. Das Zusammensetzen der 0en und 1en zu einem Codewort und das Decodieren führen die Schüler*innen jedoch von Hand durch.

Die Schüler*innen sollten bereits einfache Programme, welche die Tasten A und B als Eingabe, die LED-Matrix als Ausgabe sowie Verzweigungen und Schleifen verwenden, entworfen haben.

Erfahrungen mit dem Senden und Empfangen von Nachrichten sind hilfreich, können aber auch anhand dieser Aufgabe eingeführt werden.

Variante 3: Variante 3 sieht vor, dass die Schüler*innen ein Programm für den Calliope entwerfen, das den binären Code, der über die Tasten A und B eingegeben wird, in den entsprechenden Buchstaben übersetzt und auf der LED-Matrix ausgibt. Dazu werden im Vergleich zu Variante 2 zusätzlich algorithmische Kenntnisse im Umgang mit Zeichenketten benötigt.

Da sich hier die Motivation für den Entwurf eines binären Codes für unser Alphabet bereits aus der Decodierung durch den Calliope ergibt, ist ein Übertragen der Nachrichten an einen anderen Calliope nicht notwendig, so dass algorithmische Kenntnisse im Übertragen von Nachrichten mit dem Calliope für diese Variante nicht notwendig sind.

Lernziele

Die vorliegenden Materialien können als Einstieg in oder als Vertiefung des Moduls *Datenaustausch in Netzwerken* aus dem Lernfeld „Daten und ihre Spuren“ verwendet werden (vgl. [4], S. 16f.).

Anhand der begrenzten Eingabemöglichkeiten des Calliopes wird zunächst eine binäre Codierung unsers Alphabets motiviert. Dies ist exemplarisch für alle Codierungen zur Verarbeitung von Daten mit einem Computer. Durch die Erprobung und den Vergleich selbstentwickelter Codes, sollen die Schüler*innen nicht nur ein Verständnis dafür entwickeln, wie eine große Menge von Zeichen auf eine Darstellung mit zwei Zeichen reduziert werden kann, sondern auch dafür, welche Gestaltungsmöglichkeiten es dabei gibt. So werden einige Schüler*innen Codes fester Länge und andere Codes variabler Länge entwickeln. Letztere lassen sich nur decodieren, wenn sie präfixfrei sind oder ein definiertes Ende haben. Einige Schüler*innen entwickeln eventuell auch Codes, die zwar umsetzbar jedoch sehr ineffizient sind. Beispiele dafür sind in den Lösungshinweisen enthalten. Denkbar wäre es z. B. einen unären Code zu verwenden, der jedes Zeichen mit einer unterschiedlichen Anzahl von 0en codiert und mit einer 1 abschließt. So ein Code würde aber bereits für die 26 Buchstaben des Alphabets relativ lang.

Kommt es häufig zu Fehlern bei der Eingabe oder dem Ablesen der Codes, entsteht bei den Schüler*innen möglicherweise auch der Wunsch nach Mechanismen zur Fehlererkennung. Entweder entwickeln sie hier eigene Ansätze oder es kann eine Anregung in Richtung Paritätsbit gegeben werden.

Basierend auf den Erfahrungen mit den eigenen Codes, können dann etablierte Codes wie der ASCII-Code untersucht und seine Eigenschaften beschrieben werden. Ggf. kann auch hier ergänzend die Verwendung eines achten Bits als Paritätsbit anstelle der Verdopplung des Zeichensatzes thematisiert werden.

Werkzeuge

Hardware

Als Werkzeug bietet sich der Calliope (s. [1]) an, da er über genau zwei Eingabetasten verfügt, die eine binäre Codierung nahelegen. Zwar könnten auch die Pins zur Eingabe verwendet werden, auch damit wäre allerdings bei sechs Zeichen Schluss, wenn jede Eingabe genau einem Zeichen zugeordnet wird. Da es sich bei den Pins nicht um klassische Tasten handelt, lässt sich zudem leicht begründen, dass man sich auf die Tasten A und B beschränkt. Der Druck der Tasten A und B wird deutlich zuverlässiger erkannt, als das Berühren der Pins.

Alternativ können auch andere Physical-Computing-Werkzeuge verwendet werden, die über zwei Eingabetasten verfügen.

Software

Da es möglich sein soll, die Aufgaben mit möglichst wenig Vorkenntnissen zu bearbeiten und sich diese an Schüler*innen in der Sekundarstufe 1 richten, bietet sich eine grafische Programmiersprache an. Die Materialien wurden sowohl für die Programmierumgebung Open Roberta Lab mit der Sprache Nepo (s. [2]), als auch für Microsoft MakeCode (s. [3]) erstellt. Dabei erscheint die Umsetzung mit MakeCode in diesem Fall etwas einfacher, da ereignisgesteuert auf das Drücken der Tasten A und B und das Empfangen von Nachrichten reagiert werden kann. Dadurch wird der Programmcode kürzer und übersichtlicher.

Didaktische Hinweise zu Arbeitsblatt 1: Entwurf eigener Codes

Alle drei Varianten des Arbeitsblattes haben das Ziel, dass die Schüler*innen selbständig einen binären Code für die 26 Buchstaben des Alphabets entwickeln. Während das Codieren und Decodieren in Variante 1 und 2 von den Schüler*innen übernommen wird, wird die Decodierung in Variante 3 algorithmisch umgesetzt. Dabei ist zu beachten, dass bei dem händischen Decodieren auch Codes in Frage kommen, die nicht präfixfrei sind, wie z. B. der Morsecode, da zwischen den Zeichen eine Pause vereinbart werden kann. Bei einer algorithmischen Umsetzung wäre eine Trennung der Zeichen durch Pausen deutlich schwieriger zu erfassen. Dies kann ggf. am Beispiel des Morse-Codes mit Arbeitsblatt 3 abschließend untersucht werden.

Alle Varianten enthalten eine abschließende Aufgabe zum Vergleich der selbstentwickelten Codes mit dem ASCII-Code. Alternativ zu dieser Aufgabe, kann das zweite Arbeitsblatt verwendet werden, das den Aufbau standardisierter Codes genauer untersucht und diese als Binärzahlen interpretiert.

Variante 1

Das Arbeitsblatt ist so aufgebaut, dass zunächst ein Code entwickelt und dieser dann mithilfe der Übertragung von 0en und 1en mit dem Calliope getestet wird. Das Programm kann vorher von der Lehrkraft mit einer passenden Funkgruppe auf die Calliopes gespielt werden, so dass diese sofort einsatzbereit sind. Wichtig ist, dass sich nur je zwei Calliopes in einer Funkgruppe befinden, damit die Schüler*innen sich nicht gegenseitig beim Senden und Empfangen stören. Dieser Zugang ermöglicht

das Entwickeln und Vergleichen von Codierungen völlig unabhängig von algorithmischen Fragestellungen.

Es ist hilfreich, wenn zwei Schüler*innen gemeinsam für das Senden bzw. Empfangen zuständig sind. Beim Senden kann eine(r) den Code vorlesen und der oder die andere die Tasten drücken. Beim Empfangen kann eine(r) den Code ablesen und der oder die andere kann ihn notieren. So wird vermieden, dass Zeichen übersehen werden. Daher wird die Codetabelle hier in einer Vierergruppe entwickelt, damit sich diese anschließend in zwei Teams aufteilen und mit einer gemeinsamen Codetabelle arbeiten kann.

Nach dem Prinzip eines Gruppenpuzzles können die Gruppen anschließend neu zusammengesetzt werden, damit jeweils vier Schüler*innen mit unterschiedlichen Codetabellen zusammenkommen und ihre Codes vergleichen können.

Variante 2

Das Arbeitsblatt ist so aufgebaut, dass zunächst der Code entwickelt und getestet wird. Anschließend wird ein Programm zur Übertragung des Codes mithilfe des Calliopes entwickelt. Durch diese Reihenfolge rückt die Codierung in den Fokus und Probleme, die möglicherweise beim Entwurf des Codes entstehen, sollen aufgefangen und geklärt werden, bevor die Probleme auf der Ebene der Implementierung hinzukommen. Für Lerngruppen, die sich über die Implementierung leichter motivieren lassen, wäre aber auch die umgekehrte Reihenfolge denkbar. Tauchen beim Nachrichtenaustausch Probleme auf, muss dann jedoch genau analysiert werden, ob sich das Problem auf der Ebene der Implementierung oder der Codierung befindet. Dazu sollte die Implementierung zunächst unabhängig vom Versenden vollständiger Nachrichten getestet werden.

Die Arbeitsblätter für OpenRoberta/Nepo und MakeCode unterscheiden sich nur in Aufgabe 3.

Die Programmierung mit MakeCode ist hier ggf. etwas einfacher und intuitiver, da durch die Ereignissteuerung das Senden und das Empfangen in zwei getrennten Strängen implementiert werden können. Außerdem können die Nachrichten 0 und 1 hier als Zahlen versandt werden, da die Nachricht nur ausgewertet wird, wenn eine neue Nachricht eintrifft. Weiterhin wird die Variable, in der die empfangene Nachricht gespeichert wird, beim Verwenden des *Wenn-Datenpaket-Empfangen*-Bausteins automatisch angelegt.

In OpenRoberta hingegen werden das Drücken der Tasten und der Inhalt der empfangenen Nachricht in einer Endlosschleife immer wieder überprüft. Da die Standardbelegung einer Zahlen-Nachricht 0 ist, müssen hier die Nachrichten 0 und 1 als Zeichen verschickt werden, damit die Nachricht 0 von keiner Nachricht unterschieden werden kann. Um die Nachricht auszuwerten, muss außerdem eine Variable angelegt werden, in der sie zunächst gespeichert wird.

Je nach Kenntnisstand der Schüler*innen muss entschieden werden, ob ihnen die Tabellen mit den benötigten Bausteinen und die Programmgerüste in den Hilfe-Dateien zusätzlich zur Verfügung gestellt werden.

Variante 3

Dieses Arbeitsblatt richtet sich an etwas ältere und erfahrenere Schüler*innen. Da das zu erstellende Programm komplexer ist, wird hier zunächst mit der Codierung von nur vier Zeichen begonnen. Die damit erarbeiteten Prinzipien können dann anschließend auf eine umfangreichere Codierung des gesamten Alphabets übertragen werden. Der erstellte Code wird hier nicht nur zur Eingabe von

Zeichen verwendet, sondern der Calliope decodiert die Eingabe, um die entsprechenden Zeichen bzw. den eingegebenen Text auf der LED-Matrix auszugeben. Der Code für die Decodierung von 26 Buchstaben wird relativ lang. Den Schüler*innen wird daher angeboten sich bei der Implementierung auf einen Teil der Buchstaben, z. B. die Buchstaben ihres Namens zu beschränken.

Da sich bei dieser Variante die Motivation für den Entwurf eines binären Codes für unser Alphabet bereits aus der Decodierung durch den Calliope ergibt, kann auf das Versenden der Nachricht an einen zweiten Calliope verzichtet werden. Die Aufgabenstellung kann aber natürlich dahingehend erweitert werden, dass die Nachricht auch übertragen wird. Da es Bausteine gibt, die den Versand von Texten ermöglichen, sind hier zwei Szenarios denkbar. Entweder die Nachricht wird zunächst vollständig eingegeben und vom Sender-Calliope decodiert. Der Anwender löst dann das Versenden der gesamten Nachricht aus. Alternativ kann die Übertragung des Codes permanent wie bei Variante 1 und 2 des Arbeitsblattes erfolgen und die Decodierung erfolgt auf der Seite des Empfängers.

Je nach Kenntnisstand der Schüler*innen im Bereich Algorithmik, kann den Schüler*innen das beiliegende Grundgerüst als Hilfe zur Verfügung gestellt werden. Das Programm enthält exemplarisch die Decodierung von zwei Zeichen und ist so aufgebaut, dass es sowohl für die Decodierung von Codes fester Länge als auch für präfixfreie Codes variabler Länge verwendet werden kann.

Wird ein Code fester Länge verwendet, kann in der Endlosschleife auch zunächst überprüft werden, ob der eingegebene Code die entsprechende Länge erreicht hat. Dadurch wird der gesamte Code insbesondere für die Decodierung vieler Zeichen etwas kürzer. Der Ordner Lösungsvorschläge enthält eine entsprechende Implementierung exemplarisch für vier Zeichen.

Als Zusatz können besonders schnelle Schüler*innen auch die Darstellung der Buchstaben auf der LED-Matrix analysieren und als Code interpretiert. Dies kann auch als Überleitung zur Codierung von Bildern genutzt werden.

Didaktische Hinweise zu Arbeitsblatt 2: Standardisierte Codes

Besonders in den Blick genommen wird der ASCII-Code als Grundlage aller im Umgang mit Computersystemen gebräuchlichen Codes. Aufgrund der Erfahrungen, welche die Schüler*innen mit den Gestaltungsmöglichkeiten von Binärcodes gesammelt haben, kann der Aufbau des ASCII-Codes untersucht werden. Im weiteren Verlauf wird zum einen die Weiterentwicklung des ASCII-Codes für heutige Bedürfnisse motiviert. Zum anderen werden die Bitmuster des ASCII-Codes als Binärzahlen interpretiert. Sofern die Schüler*innen noch keine entsprechenden Vorkenntnisse besitzen, kann das binäre Zahlensystem in diesem Kontext eingeführt werden.

Die meisten ASCII-Tabellen, die im Internet zu finden sind, enthalten die Codierung der Zahlen als Dezimalzahl. Da diese Interpretation des Codes erst im Verlauf des Arbeitsblattes erarbeitet werden soll, kann für die Aufgaben die beiliegende Codetabelle des ASCII-Codes verwendet werden, die zu jedem Zeichen nur das Bitmuster enthält.

Didaktische Hinweise zu Arbeitsblatt 3: Morse-Code

Je nach Unterrichtsverlauf kann das dritte Arbeitsblatt ergänzt werden, um mit dem Morse-Code einen sehr bekannten Code aus der Zeit vor der automatisierten Datenverarbeitung zu analysieren.

Aufgrund ihrer Erfahrungen können die Schüler*innen herausarbeiten, dass sich der Code zwar zum händischen Übertragen eignet, dass die Pausen aber im Grunde einen dritten Zustand darstellen, der bei der automatisierten Decodierung schwer erfasst werden kann. Wurde Variante 1 oder 2 beim Entwurf eigener Codierungen gewählt, können die Schüler*innen die kurzen und langen Signale auch in Null und Eins übersetzen und den Calliope zur Übertragung verwenden. Bei Variante 3 müssten die Schüler*innen ein entsprechendes Programm ggf. noch implementieren oder zur Verfügung gestellt bekommen.

Ausblick

Auf Youtube findet man einen Ausschnitt der Sendung *Klein gegen Groß* vom 29.02.2020 (s. [5]). Hier ist ein Duell zu sehen, bei dem Jakob (9) behauptet: „Ich erkenne Wörter ausschließlich anhand des Binärcodes – und zwar besser als Schauspielerin Nora Tschirner.“ Dieses Duell kann entweder als Überleitung zum ASCII-Code oder als Motivation für die händische Erprobung des ASCII-Codes verwendet werden. Auch Aufgabe 5b) des zweiten Arbeitsblattes kann gut in diesem Kontext diskutiert werden.

Im Fokus der vorliegenden Materialien steht die Codierung von Zeichen. Daran anschließen kann sich eine Betrachtung der Codierung von Farben und Bildern. Insbesondere zur Untersuchung der Codierung von Farben bietet sich der Calliope ebenfalls an. Entsprechende Materialien sind im Lernfeld *Computerkompetenz* zu finden: <https://www.uni-goettingen.de/de/629169.html>

Literaturverzeichnis

- [1] Calliope gGmbH (2020). *Calliope*. <https://calliope.cc> [Datum des Zugriffs: 23.03.2020]
- [2] Das Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme. *Open Roberta Lab. Calliope mini*. Release 3.8.5 <https://lab.open-roberta.org/> [Datum des Zugriffs: 23.03.2020]
- [3] Microsoft (2018). *Microsoft MakeCode. Version 0.18.5* <https://makecode.calliope.cc> [Datum des Zugriffs: 23.03.2020]
- [4] Niedersächsisches Kultusministerium (2014). *Kerncurriculum für die Schulformen des Sekundarbereichs I Schuljahrgänge 5 – 10. Informatik*. Hannover: Unidruck
- [5] Youtube. *Klein gegen Groß* (2020). Binär-Codes entschlüsseln! IT-Experte (9) gegen Nora Tschirner | Klein gegen Groß. <https://www.youtube.com/watch?v=z3WQPXnFRBw> [Datum des Zugriffs: 09.09.2020]

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.

Die beiliegenden Materialien sind lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Sie erlaubt Bearbeitungen und Weiterverteilung des Werks unter Nennung meines Namens und unter gleichen Bedingungen, jedoch keinerlei kommerzielle Nutzung.

