**UV 6.2: Eigene Programme mit dem Calliope Mini (7-8 Ustd.)**

|  |  |
| --- | --- |
| **Inhaltsfeld**  **Inhaltliche Schwerpunkte** | **Übergeordnete Kompetenzerwartungen**  Die Schülerinnen und Schüler |
| **IF: Information und Daten**   * Informationsgehalt von Daten   **IF: Algorithmen**   * Algorithmen und algorithmische Grundkonzepte * Implementation von Algorithmen   **IF: Informatiksysteme**   * Aufbau und Funktionsweise von Informatiksystemen | **Argumentieren (A)**   * *bewerten ein Ergebnis einer informatischen Modellierung (MKR 6.4),*   **Modellieren und Implementieren (MI)**   * erstellen informatische Modelle zu gegebenen Sachverhalten, * implementieren informatische Modelle unter Verwendung algorithmischer Grundstrukturen, * überprüfen Modelle und Implementierungen,   **Darstellen und Interpretieren (DI)**   * beschreiben einfache Darstellungen von informatischen Sachverhalten, * stellen informatische Sachverhalte in geeigneter Form dar,   **Kommunizieren und Kooperieren (KK)**   * beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht, * *anstelle der vorherigen KE:* *erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht,* * kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme, * strukturieren gemeinsam eine Lösung für ein informatisches Problem, |
| **Weitere Hinweise, Vereinbarungen und Absprachen:**  Das Unterrichtsvorhaben kann unabhängig von der technischen Ausstattung der Schule durchgeführt werden. Die Anschaffung eines Mikrocontrollers ist nicht zwingend erforderlich. Alle Kompetenzen sind auch durch Nutzung der Simulationsumgebung zu erreichen.  Auf den Arbeitsblättern des Leitprogramms sind kurze Videos auf YouTube verlinkt, welche den Lerninhalt jeweils mit einem kurzen Beispiel erklären und den Schülerinnen und Schülern optional als Hilfe dienen sollen.  … zur Vernetzung:   * Bezüge zu den Grundkomponenten eines Informatiksystems aus Unterrichtsvorhaben „*Wir präsentieren uns als Avatar“* (UV 5.1) sowie zu den *Automaten in der Lebenswelt* (UV 5.4); Weiterführung der Kompetenzen aus Unterrichtsvorhaben „*Von der Anweisung zum Algorithmus“* (UV 5.2)   … zu Synergien:   * Weiterführende Projekte können im Bereich Technik die Gestaltung von Robotern, Ampeln, Messstationen usw. anregen, welche mit einem Mikrocontroller (z.B. dem Calliope Mini) gesteuert werden. | |

**Vorhabenbezogene Konkretisierung:**

Die Lernenden erkunden einen Microcontroller am Beispiel des Calliope Mini mit seinen Eingaben, der Verarbeitung und Ausgaben und ordnen seine Bauteile nach dem EVA-Prinzip zu. Dabei vergleichen Sie den Calliope Mini und seine Bestandteile mit bekannten Informatiksystemen. Anhand von einfachen Beispielen, welche beim Druck eines Buttons ein Bild oder beim Schütteln einen Smiley anzeigen, werden den SchülerInnen die Möglichkeiten des Calliope Mini (bzw. der Simulationsumgebung) verdeutlicht.

Anschließend erstellen Sie mit Hilfe der visuellen Programmiersprache MakeCode Sequenzen für Animationen auf der LED Matrix oder für die Erzeugung von eigenen Melodien. In diesem Zusammenhang werden Schleifen mit fester Dauer zur wiederholten Anzeige von Animationen oder dem wiederholten Abspielen von Melodien eingeführt.

Zur Differenzierung können Funktionen zur Modularisierung eingeführt werden, indem Programmteile in Funktionen ausgelagert werden.

Anschließend testen die Lernenden die Sensoren des Calliope Mini und nutzen diese für Verzweigungen auf Basis von Sensorwerten in Algorithmen. Dabei wird durch Kategorisierung den Sensorwerten eine Information zugeordnet (z.B. hell, dunkel, laut und leise). Hierzu erstellen die Lernenden wahlweise ein Thermometer, ein Nachtlicht oder eine Lärmampel. Dabei werden die Algorithmen durch praktisches Testen auf ihre Funktion überprüft, miteinander verglichen und anschließend zur Differenzierung als Struktogramme gesichert.

Mit Hilfe vorgegebener Programmbeispiele wird das Testen, bzw. differenzierend das Analysieren von Algorithmen vertieft. Dabei werden einzelne Programmteile auf deren Funktion und mögliche Programmfehler untersucht.

**Fokussierung der inhaltlichen Schwerpunkte:**

* Der Calliope Mini als Einplatinencomputer mit seinen Bestandteilen
* Visuelle Programmierung von Algorithmen mit den algorithmischen Grundkonzepten (Anweisungen, Sequenzen, Schleifen und Verzweigungen)
* Auf Eingaben reagieren und Sensorwerte auswerten
* Struktogramme als differenzierende Möglichkeit seine Programme zu formalisieren
* Testen bzw. Analysieren von Programmen

| **Sequenzierung:**  ***Fragestellungen*** | **Kompetenzerwartungen des Kernlehrplans**  Die Schülerinnen und Schüler | **Didaktisch-methodische Anmerkungen und Empfehlungen** |
| --- | --- | --- |
| *Kennenlernen des Calliope Mini, seiner Bauteile und der Oberfläche von MakeCode –*  *Wie kann ich ein einfaches Programm für den Calliope Mini erstellen?*  (1 Ustd.) | * benennen Grundkomponenten von (vernetzten) Informatiksystemen und beschreiben ihre Funktionen (DI) * implementieren Algorithmen in einer visuellen Programmiersprache (MI) | Kommentar:  Die Lernenden erkunden den Calliope Mini mit seinen Eingaben und Ausgaben. Anhand von einfachen Beispielen, welche bei dem Druck eines Buttons oder Schütteln etwas anzeigt, kann die grundlegende Arbeitsweise des Calliope Minis thematisiert werden. Ein kurzes Quelltextbeispiel gibt den Einstieg in die experimentelle Erkundung des Calliope Minis und dem Editor MakeCode. Die Lernenden können Ihre Programme direkt im Simulator (oder ggf. mit dem konkreten Gerät) testen.  Einstieg (Plenum):  Die Simulationsumgebung (oder ggf. das konkrete Gerät) des Calliope Mini wird gezeigt und den Schülerinnen und Schülern mit dem Aufruf eines Beispielprogramms vorgestellt. Die Schülerinnen und Schüler lesen ein Arbeitsblatt mit Beschreibungen der wichtigsten Bauteile des Calliope Mini, welche sie für ihre ersten Programme brauchen.  Vertiefung (Plenum):  Den Schülerinnen und Schülern wird das demonstrierte Beispielprogramm präsentiert und der grundlegende Aufbau des Editors erklärt.  Erarbeitung I (Einzelarbeit):  Die Schülerinnen und Schüler übertragen ein vorgegebenes Beispielprogramm und experimentieren damit. Sie testen die Programmfunktionen und verändern selbstständig Programmbausteine, um damit Veränderungen der Programmfunktionen zu erreichen. Je nach Lerngruppe können sich die Schülerinnen und Schüler hierbei auch schon mit Sitznachbarn über Ergebnisse austauschen.  Sicherung (Plenum):  Ausgewählte (abgewandelte) Programme der Schülerinnen und Schüler werden von diesen der Lerngruppe vorgestellt. Die Schülerinnen und Schüler haben die Gelegenheit eigene Programmneuerungen vorzustellen und die restliche Lerngruppe kann Nachfragen stellen.  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 1, 2) |
| Wie erstelle ich eigene Animationen und Melodien?  (1 Ustd.) | * identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI) * implementieren Algorithmen in einer visuellen Programmiersprache (MI) | Kommentar:  Die Lernenden werden an die Programmierung von Sequenzen herangeführt. Sie erstellen Befehlsabfolgen für eigene Animationen für die LED Matrix und Melodien.  Einstieg (Plenum):  Den Schülerinnen und Schülern wird der Befehlsbereich der LED-Matrix und Tonausgabe im Plenum an einem Beispiel vorgeführt.  Erarbeitung (Einzelarbeit):  Die Aufgabe besteht darin, eigene Animationen oder Melodien zu erstellen. Je nach Lerngruppe kann die Aufgabe auch darin bestehen, bekannte Melodien mit den Bausteinen nachzubilden.  Sicherung (Plenum):  Ausgewählte Animationen oder Melodien der Schülerinnen und Schüler werden der Lerngruppe vorgespielt. Bei bekannten Melodien kann jetzt die Lerngruppe versuchen, diese zu erraten.  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 3) |
| Wie kann ich wiederkehrende Muster in der Programmierung mit Hilfe einer Schleife kürzer darstellen?   1. Ustd.) | * identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI) * implementieren Algorithmen in einer visuellen Programmiersprache (MI) | Kommentar:  Die Lernenden erfahren den Nutzen von Schleifen für die Wiederholung von Animationen oder Melodien.  Einstieg (Plenum):  Die Aufgabe der vorherigen Stunde wird erneut aufgegriffen und der Lerngruppe die Programmbausteine für Wiederholungsschleifen präsentiert. Damit sollen die Schülerinnen und Schüler komplexere Animationen und / oder Melodien erstellen, bei denen sich Teile wiederholen (ohne, dass man die benötigten Befehle wiederholt notieren muss). Als Bezugsrahmen können die Schülerinnen und Schüler den Aufbau von bekannten Musikstücken beschreiben, mit Strophen und wiederkehrendem Refrain.  Erarbeitung (Einzelarbeit):  Die Schülerinnen und Schüler erweitern ihre Animationen und Melodien um weitere Teile mit Hilfe von Wiederholungsschleifen.  Sicherung (Plenum):  Ausgewählte (abgewandelte) Programme der Schülerinnen und Schüler werden von diesen der Lerngruppe vorgestellt.  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 3) |
| Sensoren und ihre Werte – Wie erhält der Calliope Mini Daten von seiner Umgebung?  (1 Ustd.) | * erläutern den Zusammenhang und die Bedeutung von Information und Daten (A), * interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI), * benennen Grundkomponenten von (vernetzten) Informatiksystemen und beschreiben ihre Funktionen (DI) * beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) als grundlegendes Prinzip der Datenverarbeitung (DI) | Kommentar:  Die Lernenden testen die Sensoren des Calliope Mini unter verschiedenen Bedingungen. Dabei wird durch Kategorisierung den Sensorwerten eine Information zugeordnet (z.B. hell, dunkel, laut und leise). Abschließend werden die Bauteile und Sensoren des Calliope Mini nach dem EVA-Prinzip zugeordnet.  Einstieg (Plenum):  Die Schülerinnen und Schüler lesen ein Arbeitsblatt mit Informationen zu den verschiedenen Sensoren des Calliope Mini. Anschließend erhalten sie die Aufgabe, mit einem Testprogramm die Sensoren unter verschiedenen Bedingungen zu testen und ihre Ergebnisse auf einem weiteren Arbeitsblatt zu notieren.  Erarbeitung I (Partnerarbeit):  Die Schülerinnen und Schüler erkunden die Sensoren im Klassenraum und notieren auf einem Arbeitsblatt jeweils die Messwerte in vorgegebenen Situationen. So lernen Sie die Bedeutung der Messwerte im realen Kontext kennen. Dabei werden die Werte Lichtstärke, Lautstärke und Temperatur gemessen. Die Lehrkraft kann zur Zeitersparnis die Sensoren auch in arbeitsteiligen Gruppen testen lassen. Bei ausschließlicher Nutzung der Simulationsumgebung, kann dieser Schritt übersprungen werden.  Sicherung I (Plenum):  Die Schülerinnen und Schüler tauschen sich über ihre Messergebnisse aus und entdecken, wo sie ähnliche Werte haben. Hierbei können auch Ungenauigkeiten der Messungen thematisiert werden.  Vertiefung (Plenum):  Das EVA-Prinzip wird noch einmal mittels eines Schaubildes wiederholt.  Erarbeitung II (Einzelarbeit / Partnerarbeit):  Die Schülerinnen und Schüler ordnen alle Bauteile und Sensoren in einer Tabelle nach dem EVA-Prinzip zu. Anschließend vergleichen Sie Ihre Zuordnungen mit einem Sitznachbarn. Schnellere Schülerinnen und Schüler ergänzen hierzu noch ihnen bekannte Informatiksysteme, in denen ähnliche Komponenten verbaut sind.  Sicherung II (Plenum):  Gemeinsames Ausfüllen der Tabelle an der Tafel auf Grundlage der Tabellen der Schülerinnen und Schüler. Sie vertiefen auf diese Weise ihr Verständnis vom EVA-Prinzip und dessen Bedeutung für Informatiksysteme.  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 4-6) |
| *Differenzierung: Wie kann man Befehle zu eigenen Befehlen zusammenfassen?*  (1 Ustd.) | * implementieren Algorithmen in einer visuellen Programmiersprache (MI) * *implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung (MI)* * *überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder ein Struktogramm (MI)* | *Kommentar:*  *Die Lernenden erstellen eigene Funktionen zur Aufteilung von Programmteilen.*  *Differenzierung: Einstieg (Plenum):*  *Den Schülerinnen und Schülern wird ein Programm mit einer Funktion demonstriert. Dabei erkennen sie, dass sich der Ablauf durch Nutzen der Funktion nicht verändert, es aber erleichtert wird, wenn man die Befehle der Funktion erneut aufrufen will, da hierfür nur noch der Funktionsaufruf nötig ist.*  *Differenzierung: Erarbeitung I (Einzelarbeit):*  *Die Schülerinnen und Schüler erstellen aus ihren bisherigen Animationen / Melodien eigene Funktionen und erweitern diese um weitere Funktionen für Animationen / Melodien. Die Funktionen können sie dann beliebig oft in ihrem Hauptprogramm aufrufen.*  *Differenzierung: Sicherung I (Plenum):*  *Ausgewählte Programme der Schülerinnen und Schüler werden von diesen der Lerngruppe vorgestellt.*  *Differenzierung: Vertiefung (Plenum):*  *Grundelemente eines Struktogrammes werden im Plenum mit Hilfe eines Arbeitsblattes besprochen.*  *Differenzierung: Sicherung II (Einzelarbeit):*  *Das eigene Animations- oder Melodieprogramm wird von den Schülerinnen und Schülern als Struktogramm notiert.*  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 7, 8) |
| Sensorwerte für Verzweigungen innerhalb von Algorithmen – Wie kann ich Entscheidungen in einem Programm umsetzen?  (1 Ustd.) | * interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI), * identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI) * implementieren Algorithmen in einer visuellen Programmiersprache (MI) * *überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder ein Struktogramm (MI)* | Kommentar:  Die Lernenden nutzen die Sensoren des Calliope Mini für Verzweigungen in Algorithmen. Dazu erstellen die Lernenden wahlweise einen Seismograph, ein Nachtlicht, ein Thermometer oder eine Lärmampel. Anschließend werden die Algorithmen durch praktisches Testen auf ihre Funktion überprüft und miteinander verglichen.  Einstieg (Plenum):  Den Schülerinnen und Schülern wird eine Lärmampel, ein Fieberthermometer und / oder ein automatisches Nachtlicht präsentiert. Sie beschreiben jeweils die Funktionsweise(n). Dabei werden die Schülerinnen und Schüler Bedingungen nennen, die z.B. bei einer Lärmampel für Grün, Gelb oder Rot sorgen. Ihnen werden daraufhin Bausteine für Verzweigungen vorgestellt, um die Messwerte für Programme nutzbar zu machen.  Erarbeitung (Partnerarbeit):  Die Schülerinnen und Schüler planen, implementieren und testen ein Programm ihrer Wahl (Seismograph, Nachtlicht, Fieberthermometer oder Lärmampel). Dabei notieren Sie ihr Vorgehen und die Funktion ihres Programms. (Zu Beginn eigenen sich der Seismograph und das Nachtlicht, da es nur zwei Zustände unterscheiden, wohingegen Lärmampel und Fieberthermometer drei Zustände beschreiben.) Schnellere Teams können mehrere Programme umsetzen. Leistungsschwächeren Gruppen können aus den Musterlösungen Quelltextpuzzle erstellt und ausgedruckt werden. Dies reduziert die Komplexität auf einen klar abgegrenzten Bereich von Befehlen, die nur noch passend angeordnet werden müssen.  Sicherung (Plenum):  Ausgewählte Programme der Schülerinnen und Schüler werden von diesen der Lerngruppe vorgestellt. Dabei können Ansätze verglichen und Probleme bei der Umsetzung besprochen werden.  *Differenzierung: Sicherung II (Einzelarbeit):*  *Das eigene Programm wird von den Schülerinnen und Schülern als Struktogramm notiert.*  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 9, 10) |
| Testen von Programmen – Was leisten die Programme und arbeiten sie korrekt?  (2 Ustd.) | * überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen (MI) | Kommentar:  Die Lernenden testen vorgegebene Programme mit Eingaben und Sensormessungen auf deren korrekte Funktion. Dies hilft die Sensibilität für korrekte Programme zu schärfen und sich darüber bewusst zu werden, dass vermeintlich intelligente Systeme immer nur so gut, wie ihre Programme sind.  Einstieg (Plenum):  Den Schülerinnen und Schülern werden verschiedene, vorgefertigte Programme präsentiert. Sie beschreiben die beobachtete Funktion der Programme.  Erarbeitung I (Partnerarbeit / Plenum):  Die Lehrkraft verteilt die vorbereiteten Programmbeispiele an die jeweiligen Teams. Das Leitprogramm stellt hierzu viele Beispiele bereit, die der Lehrkraft zur Auswahl bereitstehen. Sofern hardwareseitig Geräte vorhanden sind, können die Schülerinnen und Schüler diese in Einzel-/Partnerarbeit nutzen, um die Programme zu testen (Buttons ausprobieren, den Calliope Mini bewegen, Sensorwerte verändern durch z.B. abdunkeln usw.). Sofern mit dem Simulator gearbeitet wird, sollte dies im Plenum erfolgen, damit die Schülerinnen und Schüler nicht im Vorfeld die Programmbefehle an ihren Computern öffnen und direkt einsehen können. Neben dem zielgerichteten Testen in einer vorgegebenen Tabellenform, formulieren die Schülerinnen und Schüler in Partnerarbeit anschließend eine Funktionsbeschreibung für das jeweilige Programm und möglicher Programmfehler (ebenfalls in der Tabelle).  Sicherung I (Plenum):  Die Partnerlösungen werden im Plenum verglichen und zu einer Klassenlösung der jeweiligen Funktionsbeschreibung gebündelt. Während des Arbeitsprozesses werden die jeweiligen Testversuche auf Protokollbögen dokumentiert.  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 11- 15) |
| *Differenzierung: Analyse von Sensor-Messprogrammen und deren Bewertung –* Was leisten die Programme und arbeiten sie korrekt?  *(Differenzierung als Alternative zum „Testen von Programmen“)*  (2 Ustd.) | * *ermitteln durch die Analyse eines Algorithmus dessen Ergebnis (DI) (MKR 6.2)* * *bewerten einen als Quelltext, Programmablaufplan (PAP) oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität (A) (MKR 6.3)* | *Kommentar:*  *Zu Differenzierung kann mit Hilfe vorgegebener Beispiele zum Vergleich mehrerer Sensormessungen die Analyse und Bewertung von Algorithmen vertieft werden. Programme werden hierbei nicht nur getestet, sondern deren Programmquelltext untersucht. Die Lernenden untersuchen den Aufbau eines Algorithmus und bewerten diesen.*  *Differenzierung: Einstieg (Plenum):*  *Den Schülerinnen und Schülern werden verschiedene, vorgefertigte Programme präsentiert. Dabei geht es diesmal um die programmseitige Betrachtung des Quelltextes. Die Programme entsprechen denen aus der Vorstunde und müssen nun anhand ihres Quelltexts den Funktionsbeschreibungen zugeordnet werden. Ggf. sind hier auch noch einmal Anpassungen der Funktionsbeschreibungen notwendig.*  *Differenzierung: Erarbeitung I (Einzelarbeit):*  *Den Schülerinnen und Schülern werden arbeitsteilig Programmquelltexte auf Papier ausgeteilt, welche von ihnen analysiert werden. Die Lehrkraft verteilt die* *vorbereiteten Programmbeispiele an die jeweiligen Teams. Das Leitprogramm stellt hierzu viele Beispiele bereit, die der Lehrkraft zur Auswahl bereitstehen. Dabei werden die algorithmischen Grundstrukturen schrittweise nachvollzogen und Programmausgaben notiert.*  *Differenzierung: Erarbeitung I (Einzelarbeit):*  *Die Programmabläufe werden in Partnerarbeit verglichen und eine Funktionsbeschreibung formuliert. Diese wird anschließend bewertet, ob sie z.B. überflüssige Befehle enthält oder fehlerhaft arbeitet.*  *Differenzierung: Sicherung I (Plenum):*  *Die Teams vergleichen ihre Ergebnisse mit der Implementation, indem sie diese anschließend im Simulator testen. Während des Arbeitsprozesses werden die jeweiligen Testversuche auf Protokollbögen dokumentiert.*  (Hinweis: Zusatzmaterial „Leitprogramm“ – Seite 11-15) |

**Materialien:**

| **Nr.** | **URL / Quellenangabe** | **Kurzbeschreibung des Inhalts / der Quelle** |
| --- | --- | --- |
| 1 | https://calliope.cc | Webseite des Calliope Mini. Mit vielen grundlegenden Hinweisen und weiterführenden Links zu Materialien. |
| 2 | https://makecode.calliope.cc | Der MakeCode Editor von Microsoft. Ein visueller, blockbasierter Editor für den Calliope Mini mit Simulationsumgebung. |

letzter Zugriff auf die URLs: 31.01.2022

*[Diese Liste/Diese Veröffentlichung/Dieses Angebot enthält Links zu externen Websites Dritter, auf deren Inhalte QUA-LiS NRW keinen Einfluss hat. Dementsprechend obliegt die Einhaltung der datenschutzrechtlichen Regelungen dem jeweiligen Anbieter bzw. Betreiber. Im Sinne der gesetzlichen Gesamtverantwortung für den Datenschutz an Schulen prüfen Schulleitungen daher vor einem Einsatz der genannten Quellen eigenverantwortlich, inwieweit und unter welchen Bedingungen die Nutzung der genannten Quellen für den beabsichtigten Zweck datenschutzrechtskonform möglich ist. Ggf. resultiert aus einer solchen Prüfung im konkreten Fall, dass die allgemeine Nutzung weitestgehend nur auf freiwilliger Basis möglich ist, d.h. Schülerinnen und Schüler (oder deren Erziehungsberechtige) bzw. Lehrerinnen und Lehrer nicht oder nur eingeschränkt zur Nutzung verpflichtet werden können.]*