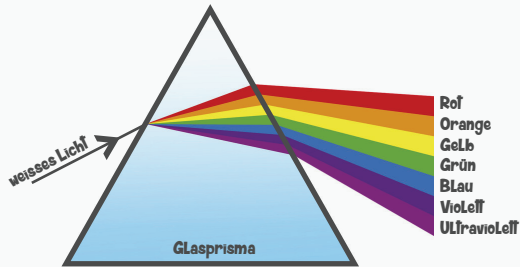


Farben

Ohne Licht gäbe es keine Farben

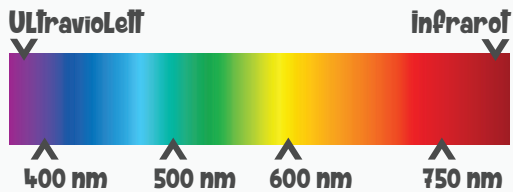
Ohne Licht gäbe es keine Farben auf der Welt. Lässt man weisses Licht durch ein Glasprisma fallen, sieht man, dass sich das Licht in dem Glasprisma bricht und in verschiedene Farben aufspaltet. Denn Licht besteht aus speziellen Wellen, wobei jede Farbe eine andere Wellenlänge hat und im Prisma unterschiedlich stark gekrümmt wird.



Das bedeutet: Weisses Licht ist aus farbigen Lichtern zusammengesetzt. Diese sieht man auch bei einem Regenbogen.

Wie sehen wir Farben?

Das Auge reagiert auf optische Reize und kann Licht mit bestimmten Wellenlängen wahrnehmen. Wir Menschen können Licht nur in einem sehr schmalen Wellenlängenbereich wahrnehmen. Er reicht von Violett (400 nm) bis Rot (750 nm).

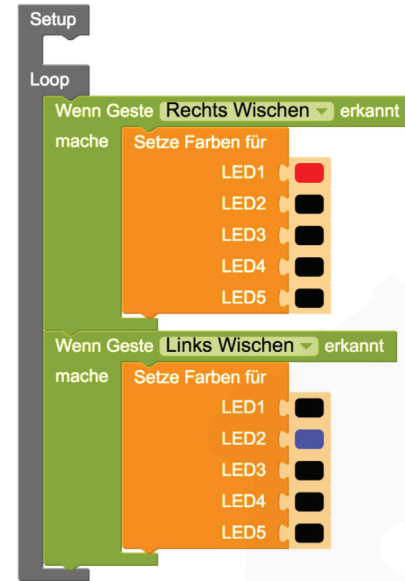


Unsere Augen nehmen über 200 verschiedene Farbtöne wahr, unterscheiden ganz selbstverständlich auch zwischen feinsten Nuancen, kennen über 20 Sättigungs- und 500 Helligkeitsstufen. Verantwortlich sind zwei verschiedene Arten von Sinneszellen: Stäbchen und Zapfen. Die Stäbchen lassen uns die Helligkeitsunterschiede bis zu einer bestimmten Lichtintensität wahrnehmen. Die Aufgabe der Zapfen ist die Farbwahrnehmung. Sie existieren in verschiedenen Varianten, die jeweils auf unterschiedliche Wellenlängen des Lichts reagieren.

Aufgabe A

1. Farben programmieren

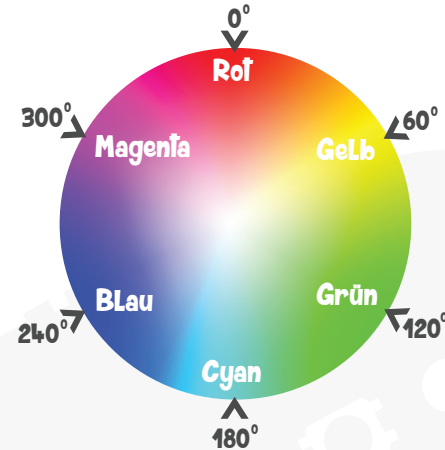
Das EDU-Board erkennt Gesten, wie Links- und Rechtswischen. Erstelle ein Programm, das eine LED in verschiedenen Farben bei Links- oder Rechtswischen ein- und ausschaltet.



Das EDU-Board kennt noch mehr Gesten. Entwickle nun deinen eigenen Farben-Code.

2. Farben erkennen: vom Lichtspektrum zum Farbkreis

Das für uns sichtbare Lichtspektrum von Rot bis Violett kann auch als Farbkreis dargestellt werden. In der Farbenlehre kann so eine Farbe nach ihrem Farbton, ihrer Helligkeit und ihrer Sättigung beschrieben werden. Diese Begriffe wurden auch bei der Programmierung des EDU-Boards benutzt. Der Farbkreis wird im Uhrzeigersinn gelesen und reicht von 0 bis 360 Grad. So lässt sich beispielsweise der Farbwert einer Farbe ermitteln.



3. Zeig mir die Farbe – ich zeige dir den Farbwert!

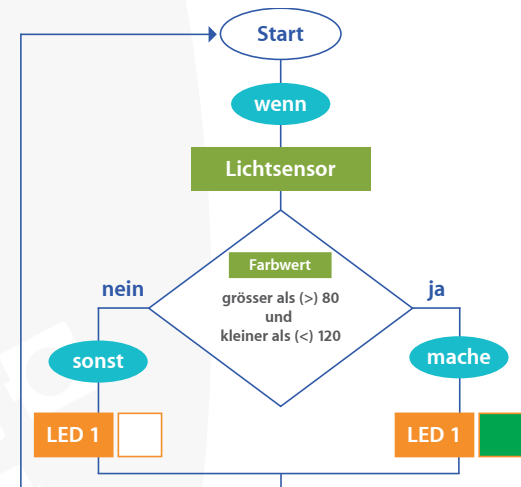
Welche Farbwerte die verschiedenen Farben haben, kann dein EDU-Board erkennen. Da es ohne Licht keine Farben gibt, ist der erste Programmierschritt folgender: Schalte Status LED ein.

Ermittle die Werte weiterer Farben.

: hue:
: hue:
: hue:

Mit den gemessenen Farbwerten kannst du nun ein Programm entwickeln, das Farben eines bestimmten Farbspektrums erkennt.

Das folgende Flussdiagramm zeigt dir, wie du vorgehen kannst.



4. Wie funktioniert ein Scanner?

Ein Scanner tastet Vorlagen mit lichtempfindlichen Sensoren ab und wandelt die daraus gewonnenen Signale in digitale Daten um.

5. Und mein Smartphone?

Der Lichtsensor im Smartphone dient dazu, die Helligkeit des Bildschirms an die gegenwärtige Helligkeit des Raumes anzupassen. In modernen Smartphones sind drei einzelne Lichtsensoren für die Farben Rot, Grün und Blau verbaut. Mit einem solchen Farbsensor kann neben der Helligkeit auch die Farbsättigung des Displays an die Umgebungsbeleuchtung angepasst werden.

Gelb: hue:
Rot: hue:
Grün: hue:

Lichtsensord

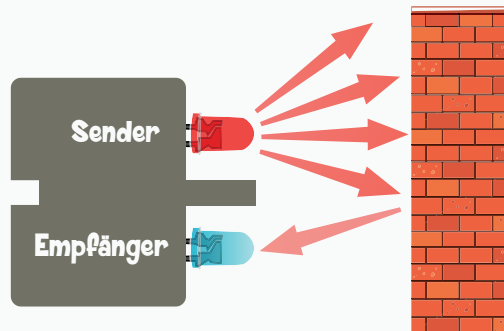
Wie können wir Distanzen wahrnehmen?

Unsere beiden Augen liefern ein leicht verschobenes Bild aufgrund ihrer unterschiedlichen Perspektiven. Das Gehirn gleicht diese beiden Bilder mit



Gelerntem und Erfahrenem ab und erzeugt ein dreidimensionales Bild. Das erlaubt uns, Distanzen abzuschätzen und räumlich zu sehen. Trotzdem fällt es uns in einigen Situationen schwer, Distanzen richtig einzuschätzen. Im Alltag nehmen uns Sensoren diese Aufgabe ab. So haben z.B. die meisten Autos eine Sensorvorrichtung mit Anzeige, die uns das Einparken erleichtert.

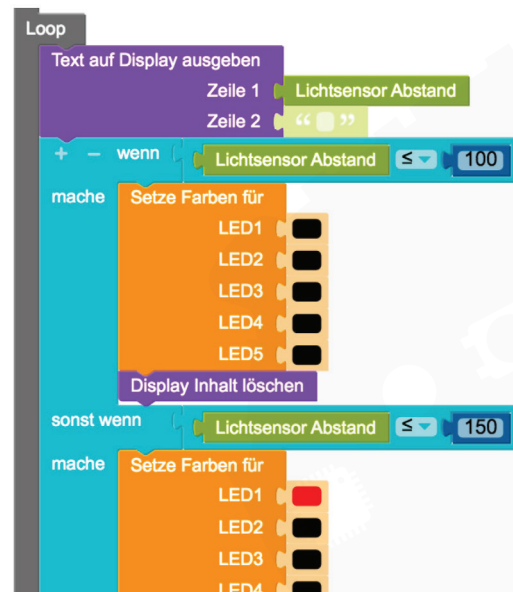
Der Lichtsensor kann Helligkeitsunterschiede erkennen. Dabei sendet eine kleine LED Licht aus, das von der zu messenden Oberfläche reflektiert und gemessen wird. So können z.B. Entfernungen gemessen werden.



Aufgabe B

6. Parksystem programmieren

Ein Parksystem zeigt z.B. über eine LED-Anzeige die Annäherung an ein Hindernis an und warnt bei sehr geringem Abstand (z.B. 15 cm oder weniger). Programme nun ein solches Parksensorsystem mithilfe des Lichtsensors. Das EDU-Board soll bei Annäherung eines Gegenstandes zuerst eine, dann zwei und schließlich drei LEDs aufleuchten lassen.



Entwickle dein Programm weiter, indem du neben den LEDs zusätzlich auch Signaltöne oder Texte im Display programmierst.



7. Personenzähler

Kleidergeschäfte nutzen Sensoren, um das Einkaufsverhalten der Kunden zu analysieren. Ein sogenannter Kundenzähler zählt die Personen, die ein Geschäft betreten und wieder verlassen. Das geschieht über einen Lichtsensor am Eingang des Geschäfts. Betritt ein Kunde das Geschäft, wird eine Lichtschranke unterbrochen, ein Signal weitergeleitet und ein Zählvorgang ausgelöst.

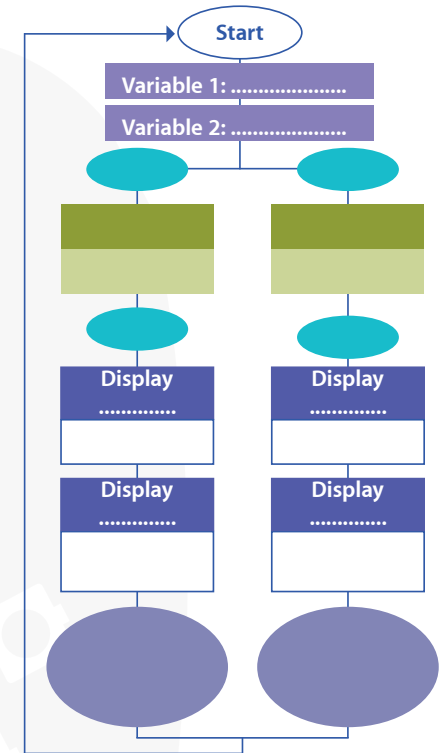
8. Kundenzähler programmieren

Programmiere mit den vorhandenen Sensoren auf dem EDU-Board einen Personenzähler. Teste das Programm am Eingang des Schulzimmers. Das Programm soll unterscheiden zwischen Personen, die das Zimmer betreten, und Personen, die das Zimmer wieder verlassen.



9. Ist dein Programm alltagstauglich?

Entwickle das Programm mit Hilfe des Flussdiagrammes weiter.



10. Weitergedacht

Wo finden Lichtsensoren noch Verwendung? Finde Beispiele aus deinem Alltag und erkläre die Funktionsweise.

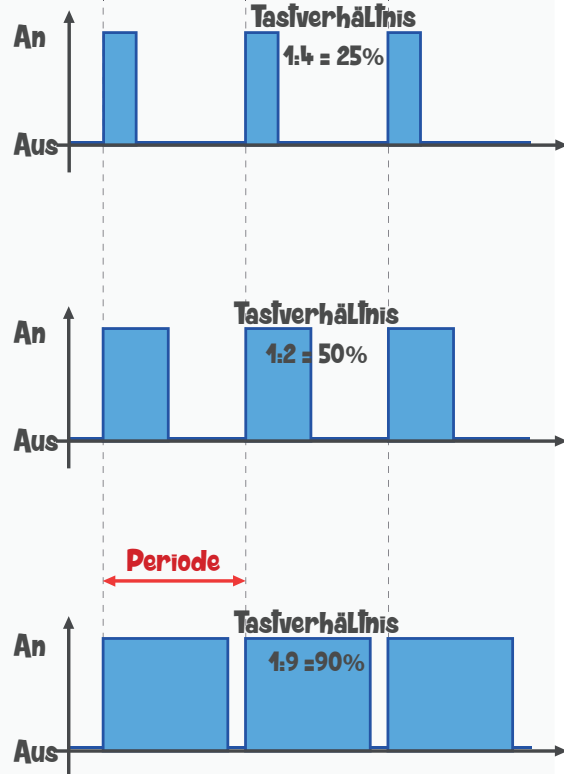
LED

Was ist eine LED?

LED, die Licht emittierende Diode, bezeichnet ein elektronisches Bauelement. Fließt durch die Diode Strom in Durchlassrichtung, so strahlt sie Licht ab. Im Gegensatz zu herkömmlichen Leuchtmitteln sind LEDs elektronische Bauteile: winzige Elektronik-Chips aus Halbleiterkristallen.



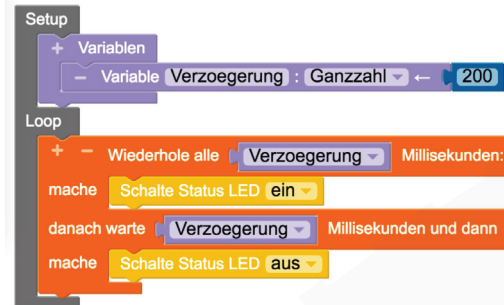
LEDs lassen sich auch dimmen, denn sie werden oft mit gepulstem Strom betrieben. Das heißt, der Strom wird so gesteuert, dass er dauernd an und wieder aus geht. Um die Helligkeit steuern zu können, wird der Strom in schneller Folge ein- und ausgeschaltet, was unser träges Auge als Dimmung wahrnimmt. Man nennt dieses Verfahren auch Pulsweitenmodulation (kurz: PWM).



Aufgabe C

11. Status-LED programmieren

Programmiere mit der Status-LED ein Blinklicht, das mit einer Verzögerung an und aus geht. Um einen Befehl dauernd auszuführen, benötigen wir eine Schleife. Wiederholt werden soll unsere Variable, die wir Verzögerung nennen.

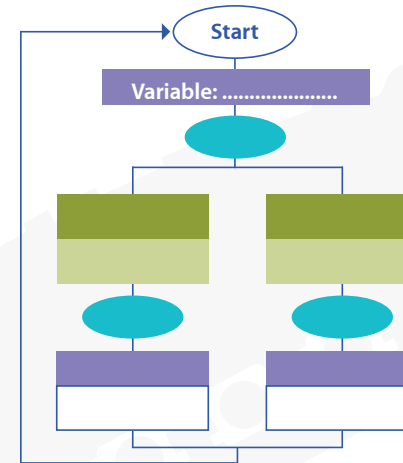


Mit diesem Wert ist die LED 200 ms eingeschaltet und danach 200 ms ausgeschaltet. Teste Werte der Variable zwischen 1 und 500 ms. Was stellst du fest?

12. Programmerweiterung

Bis jetzt haben wir die An- und Ausschaltzeit durch das Eintippen einer Zahl verändert. Das EDU-Board kann aber noch mehr. Ähnlich wie bei einem Gamecontroller, besitzt es Eingabetasten. Erweitere das Programm, in dem du mit den Tasten die Ein- und Ausschaltzeit des LEDs variieren kannst.

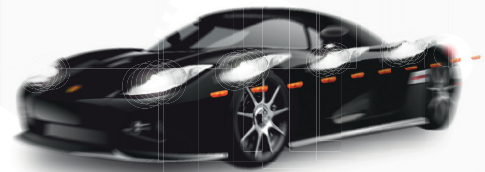
13. Plane dein Vorgehen wieder mit einem Flussdiagramm



Bei welchen Werten kannst du die LED noch blinken oder flimmern sehen? Finde die Grenzwerte und vergleiche diese Werte mit deinem Partner.

14. Perlschnureffekt

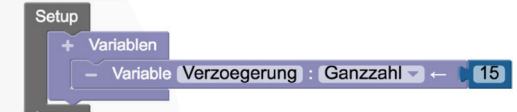
Als Perlschnureffekt bezeichnet man eine optische Täuschung, verursacht durch rasche Augenbewegungen auf eine gepulst betriebene Lichtquelle hin oder von dieser weg. Aufgrund der Trägheit



unserer Augen, führt es zum mehrfachen Erscheinen der Lichtquelle, z. B. bei PWM-gediminten LED-Verkehrstafeln oder LED-Autorückleuchten. Ist dir dieses Phänomen auch schon mal aufgefallen?

15. Entwickle dein Programm

Halte dein EDU-Board mit dem bereits erstellten Programm und eingeschalteter LED vor dich. Bewege es ganz schnell vor dir hin und her. Was kannst du sehen? Untersuche verschiedene Werte zwischen 1 und 15 ms. Zeichne deine Beobachtungen auf und erkläre sie.



16. Entwickle dein Programm weiter

Was musst du mit der Variable tun, um die LED zu dimmen? Tipp: Beim Perlschnureffekt hast du gesehen, wie lange die LED an und aus war.

17. Recherche

Recherchiere, wie in der Digitaltechnik ein digitales Signal gezeichnet wird. Kannst du deine Zeichnungen vom Perlschnureffekt auch als digitales Signal darstellen und beschreiben?

Alarm- und Überwachungssystem

Ausgangslage

Du bist Mitarbeiter einer Sicherheitsfirma, die individuelle Sicherheitssysteme plant und realisiert. Nun wirst du mit einem neuen Auftrag vertraut gemacht. Erstelle ein Programm, das die folgenden Sicherheitskomponenten enthält. Das EDU-Board soll das Sicherheitssystem simulieren. Dir stehen alle Sensoren des EDU-Boards zur Verfügung. **Signalöne sind wegen Prüfungssituation nicht erlaubt.**



A: Einbruchmeldeanlage

Bei unbefugtem Betreten des Gebäudes soll ein Alarm ausgelöst werden. Das Alarmsystem soll mithilfe einer Taste wieder ausgeschaltet werden können.



Test

B: Temperaturüberwachung



Im Lagerraum des Auftraggebers befinden sich sensible Gerätschaften. Der Lagerraum soll eine optimale Temperatur von 22 – 26°C aufweisen. Das Display soll die aktuelle Temperatur anzeigen. Zudem soll ein Warnhinweis erscheinen, wenn die optimale Temperatur unter- oder überschritten wird.

C: Zutrittskontrolle

Die Mitarbeiter des Auftraggebers sollen über einen Sicherheitscode Zutritt ins Gebäude haben. Programmiere einen vierstelligen Sicherheitscode mithilfe der Tasten oder der Gestenfunktionen. Das Display soll anzeigen, ob die Code-Eingabe korrekt war. Nach 3 Sekunden soll sich das Schloss wieder selbständig schliessen und eine neue Codeeingabe einfordern.

Teste dein Programm auf die Alltagstauglichkeit. Nenne Verbesserungsmöglichkeiten und erkläre, wie du vorgehen würdest.

Alltagstauglichkeit, Verbesserungsmöglichkeiten:

Persönliche Angaben

Klasse:

Geschlecht: ☐ W ☐ M

EDU-Board Seriennummer:

Vorname Vater (1. Buchstabe):

Vorname Mutter (1. Buchstabe):

Geburtsdag (09.10.2004 Bsp. 09):

Bewertung Test (Programm A, B, C)

Teilprogramm A

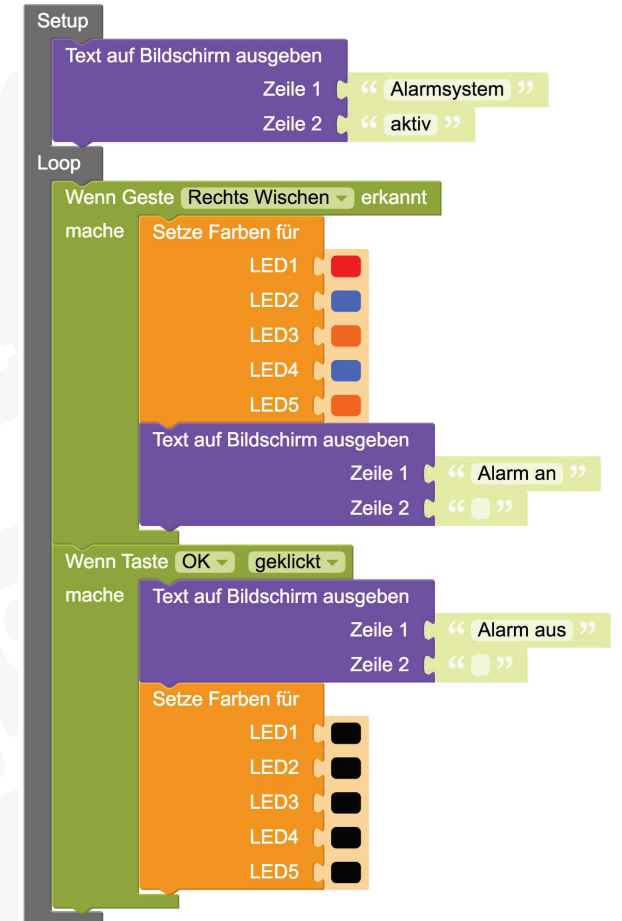
Funktionalität				Erreichte Punktzahl
	0	1	2	
	Programm funktioniert nicht, mehr als ein Fehler eingebaut	Programm funktioniert nicht, maximal ein Fehler eingebaut	Programm funktioniert einwandfrei	

Komplexität				Erreichte Punktzahl
Eingabe	0	1	2	
	Kein Sensor verwendet	Ein Sensor verwendet (z.B. Gesten- Sensor für Alarm-aktivierung und Deaktivierung)	Mehrere Sensoren sinnvoll verwendet (z.B. Gesten-Sensor für Alarmaktivierung, Druckknopf für Deaktivierung)	
Ausgabe	0	1	2	
	Keine Ausgabe	Display oder LED verwendet	Display und LED verwendet	

Total Erreichte Punktzahl (Teilprogramm A)

Max. 6 Punkte

Musterlösung Einbruchmeldeanlage



Bewertung Test (Programm A, B, C)

Teilprogramm B

Funktionalität				Erreichte Punktzahl
	0	1	2	
	Programm funktioniert nicht, mehr als ein Fehler eingebaut	Programm funktioniert nicht, maximal ein Fehler eingebaut	Programm funktioniert einwandfrei	

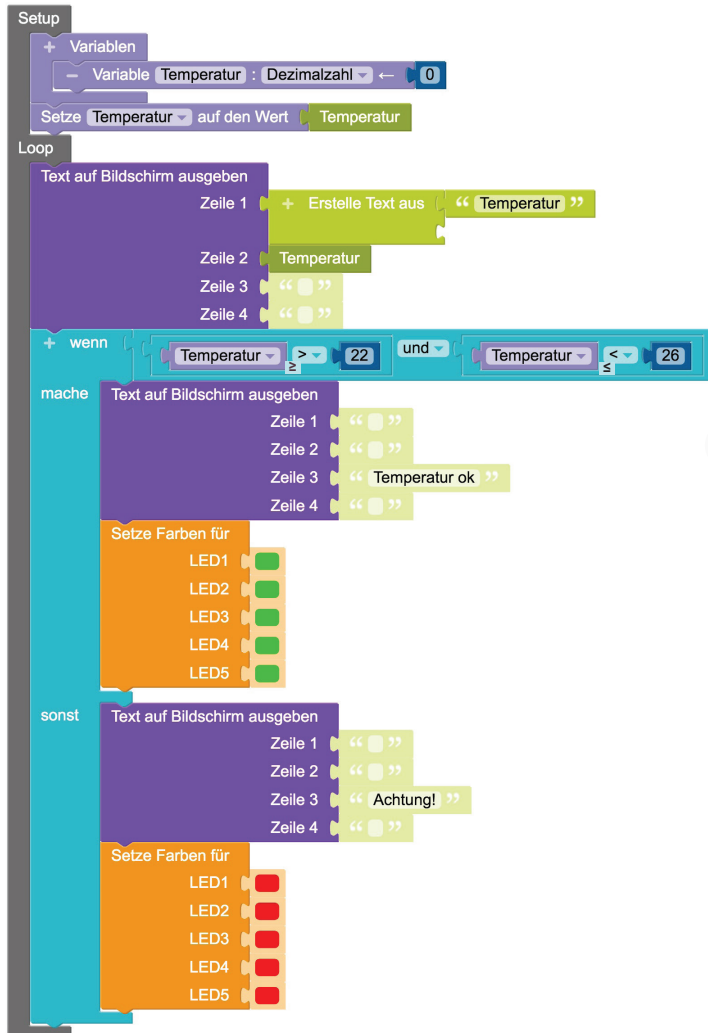
Komplexität				Erreichte Punktzahl
Eingabe	0	1	2	
	Keine Variable verwendet	Variable eingesetzt	Variable eingesetzt und richtig verknüpft	
Verarbeitung	0	1	2	
	Kein Sensor verwendet	Temperatursensor für untere oder obere Temperaturgrenze richtig verwendet (wenn <23 oder ≤ 22) (wenn >25 oder ≥ 26)	Temperatursensor für untere und obere Temperaturgrenze richtig verwendet (wenn <23 oder ≤ 22) (wenn >25 oder ≥ 26)	
Ausgabe	0	1	2	
	Keine Ausgabe	Display oder LED sinnvoll verwendet	Display und LED sinnvoll verwendet	

Total Erreichte Punktzahl (Teilprogramm B)	Max. 8 Punkte
---	---------------

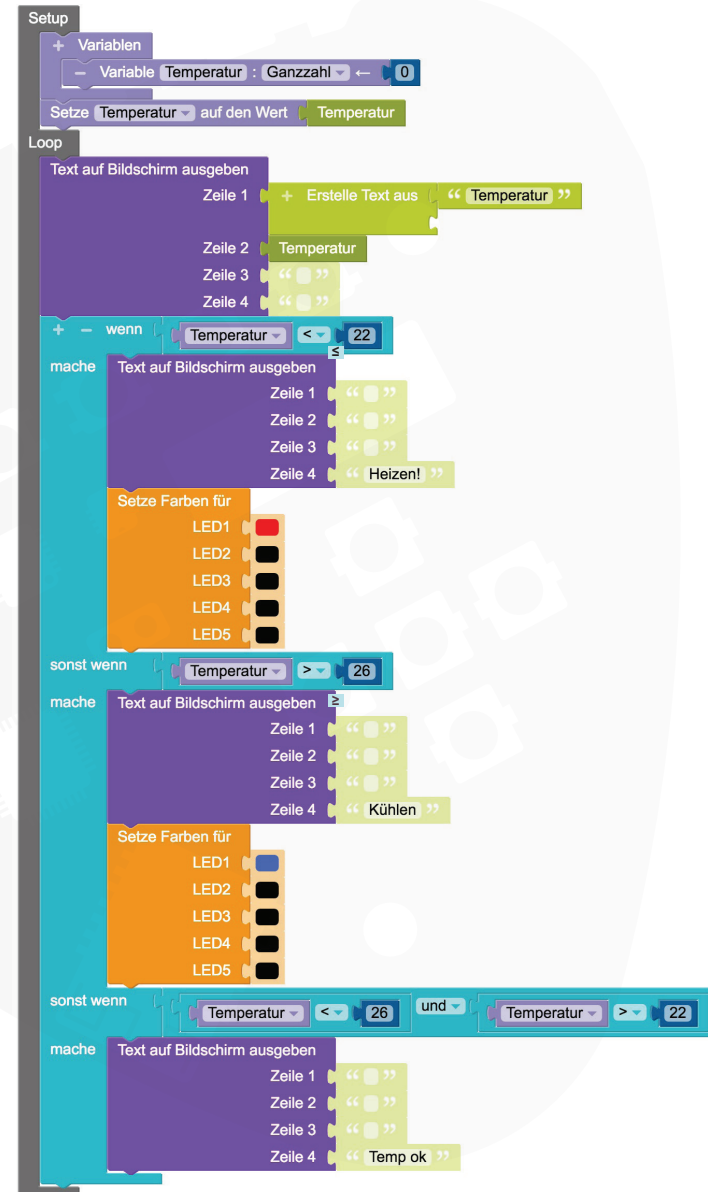
Bewertung Test (Programm A, B, C)

Teilprogramm B: MusterLösung Temperaturüberwachung

(mit Logik wenn-mache-sonst)



(mit Logik wenn-mache)



Bewertung Test (Programm A, B, C)

Teilprogramm C

Funktionalität				Erreichte Punktzahl
	0	1	2	
	Programm funktioniert nicht, mehr als ein Fehler eingebaut	Programm funktioniert nicht, maximal ein Fehler eingebaut	Programm funktioniert einwandfrei	

Komplexität				Erreichte Punktzahl
Eingabe	0	1	2	
	Keine Variable verwendet	Variable eingesetzt	Variable eingesetzt und richtig verknüpft	
	0	1	2	
	Kein logischer Wert verwendet	Logischer Wert verwendet	Logischer Wert verwendet und richtig verknüpft	
	0	1	2	
	Kein Sensor zur Codeeingabe verwendet	Programm als einstelliger Code verwendbar	Programm als mehrstelliger Code verwendbar	
Verarbeitung	0	1	2	
	Keine vollständige Codeeingabe möglich	Eine vollständige Codeeingabe möglich (Programm wird nach erster Codeeingabe nicht zurückgesetzt)	Mehrere vollständige Codeeingaben möglich (Programm wird nach erster Codeeingabe wieder zurückgesetzt)	
Ausgabe	0	1	2	
	Keine Ausgabe	Display mehrfach verwendet (z.B. „Code eingeben“, Display löschen, „offen“)	Display und LED mehrfach verwendet	

Modellkritik				Erreichte Punktzahl
	0	1	2	
	Keine kritische Auseinandersetzung mit dem erstellten Programm	Kritische Auseinandersetzung mit dem erstellten Programm, aber ohne Ursachenerkennung, kein Vorschlag zur Programmverbesserung	Kritische Auseinandersetzung mit dem erstellten Programm, mit Ursachen-erkennung und nachvollziehbarer Vorschlag zur Programmverbesserung	

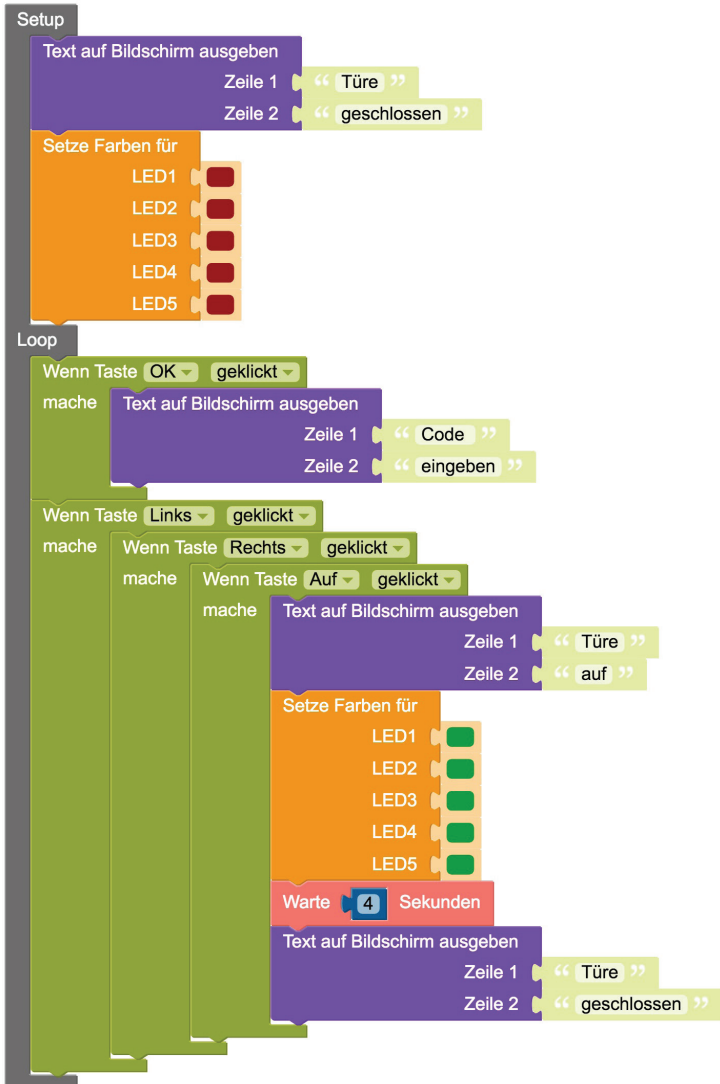
Total Erreichte Punktzahl (Teilprogramm C)

Max. 14 Punkte

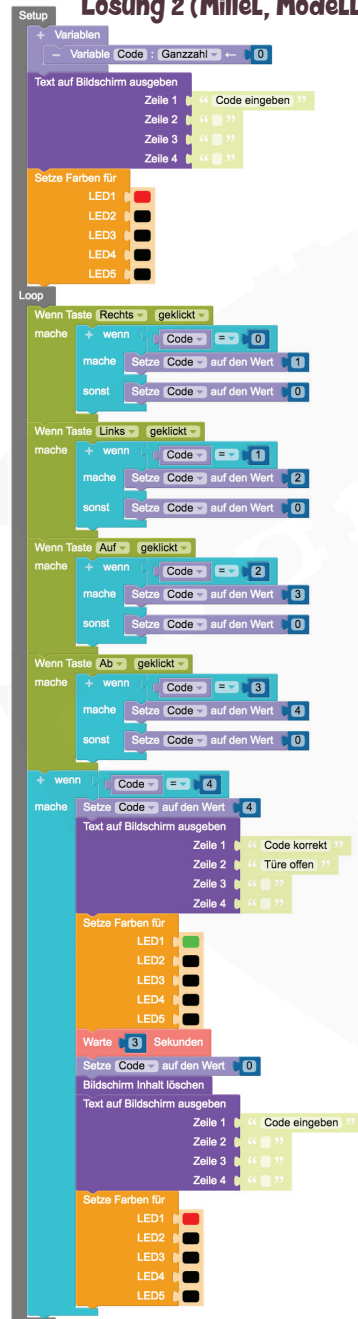
Bewertung Test (Programm A, B, C)

Teilprogramm C: MusterLösung Zutrittskontrolle

Lösung 1 (Einfach, hohe Modellkritik)



Lösung 2 (Mittel, Modellkritik)



Lösung 3 (Schwer)

