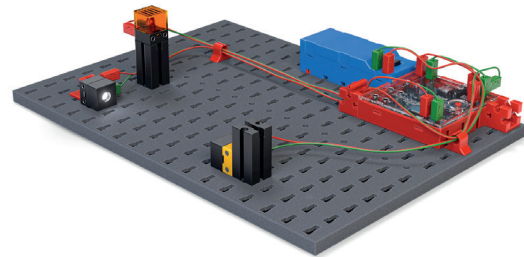


Modell 7: Telegraf – Signale übertragen & codieren

Worum geht es? Signale durch die Luft zu übertragen war lange nur ein Traum der Menschheit. Als es endlich möglich war, hat diese Technologie die Entwicklung von vielen Dingen, z.B. des Telefons erst möglich gemacht. Lerne selbst die ersten Schritte der Signalübertragung kennen!



Einführung

Eine der bedeutendsten Erfindungen des 19. Jahrhunderts ist die **Telegrafie** – die schnelle Übertragung von Nachrichten über große Entfernungen.

Heute ist es für uns selbstverständlich, dass wir mit anderen Menschen überall auf der Welt telefonieren können, als ob sie neben uns stünden. Das war aber nicht immer so. Lange waren Boten und die Post die einzige Möglichkeit, um Nachrichten über größere Entfernungen zu übermitteln.

Da konnte es schonmal ein paar Tage dauern, bis eine Nachricht bei einem Empfänger ankam.

Mit dem elektrischen Strom wurde dann eine Übertragung mit Lichtgeschwindigkeit möglich.

Mit diesem Modell wirst du einen einfachen **Telegrafen** kennenlernen, der mit Licht funktioniert.

Bei diesem Modell wirst du die Zeit messen müssen. Das geht in Scratch sehr einfach mit der schon vorhandenen Variablen **„Stoppuhr“**.

Diese Variable zeigt die Sekunden seit dem Programmstart oder dem letzten Zurücksetzen der Uhr:



1 Zeit anzeigen + "Stoppuhr" zurücksetzen

Setze nun das Häkchen an der Stoppuhr-Variablen (im Menu „fühlen“).

Dann wird die Zeit in Sekunden auf der „Bühne“ angezeigt.

Mit dem folgenden Scratch-Befehl kannst du die Stoppuhr auf „0“ zurückstellen. Die startet anschließend sofort wieder bei 0 Sekunden.



2 Konstruktionsaufgabe

Baue den Telegrafen nach der Bauanleitung auf: Wenn ihr zwei ft-Controller zur Verfügung habt, dann baut den Sender und den Empfänger getrennt auf. Andernfalls baut ihr den Telegrafen ("Morsestation"), die sowohl einen Sender als auch einen Empfänger enthält.

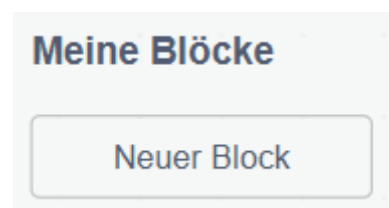
Modell 7: Telegraf – Signale übertragen & codieren

Programmierbefehle, die sich oft wiederholen? Spare Zeit mit "Blöcken"

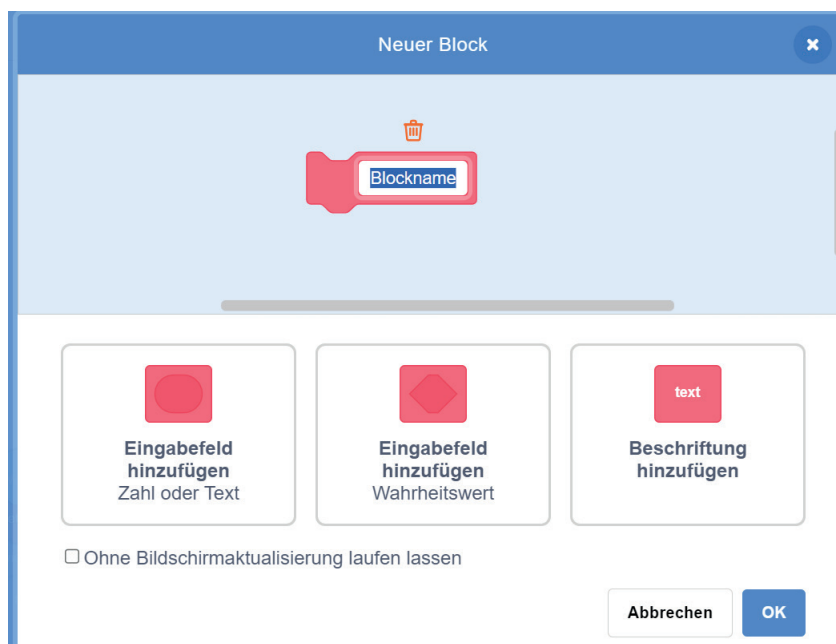
Bei den Programmen in diesem Aufgabenblatt werden sich öfter Befehlsfolgen **wiederholen**.

Um deine Scratch-Programme übersichtlicher zu machen, bietet dir Scratch die Möglichkeit, aus solchen Befehlsfolgen eigene Blöcke („Unterprogramme“) zu machen, die du dann wie einen Scratch-Befehl aufrufen kannst.

Um einen solchen eigenen Block zu erschaffen musst du auf die **Schaltfläche** „**Neuer Block**“ klicken:



Und dann einen eigenen Namen für dein Unterprogramm eingeben. Dieser Name steht nun auf deinem **neuen Block**. Du kannst ihn nun wie einen Scratch-Befehl in deine Programme einbauen.



Programmieraufgaben

1 Lichtsignal

Der Telegraf funktioniert ganz ähnlich wie die Lichtschranke, die du in bereits bei der Alarmanlage kennengelernt hast: Die **LED** bildet den Sender und der **Fototransistor** den Empfänger.

Der Sender schickt dabei Lichtzeichen, und der Empfänger erkennt, wenn das Licht auf den Fototransistor fällt.

Starte nun als Sender-Programm das Blinklicht-Programm aus dem ersten Modell.

Hinweis: Wenn ihr mit getrenntem Sender und Empfänger arbeitet, sollte der Sender neben der Sende-LED auch die rote LED ein- bzw. ausschalten, damit ihr erkennt, dass ein Signal gesendet wird.

Ergänzt das Programm entsprechend.

Speichere das Programm unter dem Namen „**Sender**“.

Programmiere ein Scratch-Programm für den Empfänger, das darauf wartet, dass der Fototransistor ein Licht-Signal des Senders erkennt.

Dann soll es die rote Empfangs-LED einschalten, und sie wieder ausschalten, sobald kein Lichtsignal vom Sender mehr erkannt wird.

Hinweis: Wenn ihr mit getrenntem Sender und Empfänger arbeitet, dann **testet** das Programm des Empfängers, indem ihr das Programm „Sender“ beim Sender startet.

Speichere das Empfänger-Programm unter dem Namen „**Empfänger**“.

An der Morsestation musst du das „Sender“- (Blinklicht-) und das „Empfänger“-Programm gleichzeitig starten. Wie das geht, hast du beim Ventilator gelernt.

Speichere das kombinierte Sende- und Empfangs-Programm unter dem Namen „**Sender - Empfänger**“.

Programmieraufgaben

2 Übermittlung einer Zahl

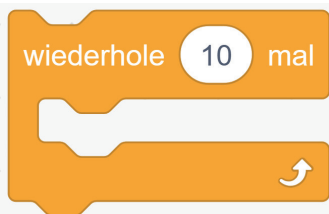
Dein Telegraf kann bisher nur ein einzelnes Signal („an“, „aus“) senden. Für eine Nachricht ist das sehr wenig Information.

Jetzt wollen wir versuchen, mehr Informationen zu übermitteln: Der Sender soll Zahlen an den Empfänger übermitteln. Dazu soll der Sender mehrmals blinken und der Empfänger die gesendeten „an“-Signale zählen.

Programm zum Senden:

Schreibe zunächst das Sende-Programm aus der vorherigen Aufgabe so um, dass es eine vorgegebene Anzahl „an“-Signale sendet.

Dafür kannst du den Scratch-Befehl "wiederhole ... mal" verwenden:



Speichere es unter dem Namen „Zahl-Sender“.

Programm zum Empfangen:

Der Empfänger soll die Anzahl der empfangenen Signale in einer Variablen „Zahl“ speichern.

Wie das geht, hast du beim Modell „Münz-zähler“ gelernt: Das Scratch-Programm des Empfängers muss dazu bei jedem neuen „an“-Signal die Variable „Zahl“ um eins erhöhen.

Schreibe dein bisheriges Empfänger-Programm so um, dass es die Signale des Senders zählt.

Tipp: Nachdem du den Wert der Variablen „Zahl“ um eins erhöht hast, musst du warten, bis das Signal des Senders wieder „aus“ ist, damit du kein Signal mehrfach zählst.

Teste das Programm zusammen mit dem Programm des Senders und **speichere** es unter dem Namen „Zahl-Empfänger“.

Hinweis: An der „Morsestation“ müsst ihr, wie in der vorherigen Aufgabe, das Sende- und das Empfangs-Programm zusammen starten und speichern.

Programmieraufgaben

3 Zahlencode-Sender

Mit dem Programm aus Aufgabe 2 kannst du nun genau **eine** Zahl übertragen. Für eine **ganze Nachricht** ist das allerdings noch immer nicht viel.

Mit einer kleinen **Ergänzung** deines Send-Programms kannst du nun eine Folge von Zahlen übermitteln.

Damit der Empfänger weiß, wann eine Zahl fertig übermittelt ist und eine neue folgt, soll der Sender nach jeder Zahl eine größere Pause von zwei Sekunden machen.

Ergänze zunächst dein Sender-Programm aus Aufgabe 2, so dass es mehrere Zahlen nacheinander übermittelt.

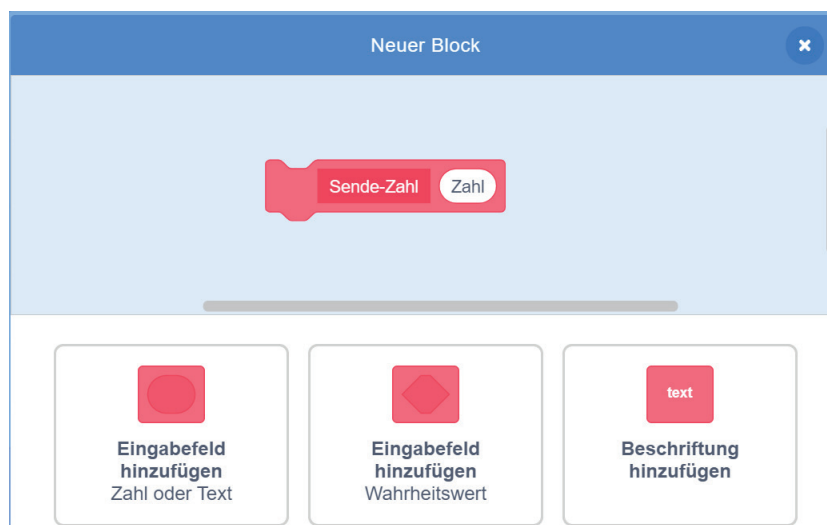
Damit dein Scratch-Programm übersichtlicher wird, sollst du nun einen eigenen Block „**Sende-Zahl**“ definieren, der die Übermittlung einer Zahl übernimmt.

So geht es: du machst in Scratch einen eigenen Block, der eine entsprechende Anzahl „an“-Signale übermittelt und danach zwei Sekunden Pause macht.

Bei der Erschaffung des Blocks musst du auf „Eingabefeld hinzufügen“ klicken und für den Block das **Eingabefeld „Zahl“** festlegen.

In deinem Sender-Programm kannst du nun den Block wie einen Scratch-Befehl mehrmals hintereinander aufrufen, um verschiedene Zahlen zu senden.

Speichere das Programm unter dem Namen „**Zahlencode-Sender**“.



Programmieraufgaben

4 **Zahlencode-Empfänger**

Das zu dem Zahlencode-Sender passende Empfänger-Programm muss die empfangenen „an“-Signale zählen.

Auch dabei sollst du einen eigenen **Block („Empfange-Zahl“)** erschaffen und programmieren. Die empfangene Zahl soll auf der „Bühne“ angezeigt werden.

Tipp: Nach jedem Signal muss der Block „Empfange-Zahl“ die Stoppuhr starten und prüfen, ob es mehr als 1.5 Sekunden dauert, bis das nächste Signal kommt.

Verwende dafür die Stoppuhr.

Erweitere dein Scratch-Programm aus Aufgabe **2** entsprechend.

Teste es mit dem Zahlencode-Sender aus der vorherigen Aufgabe und **speichere** es unter dem Namen **„Zahlencode-Empfänger“**.

Hinweis:

Wenn du die „Morsestation“ verwendest, musst du auch hier beide Programme zusammen starten und speichern.

Experimentieraufgabe

5 Nachrichtentext

Mit deinem Zahlencode-Telegrafen kannst du nun ganze Nachrichten übermitteln. Nummeriere dazu die Buchstaben des Alphabets durch („A“ = 1, „B“ = 2, ...):

A	
B	
C	
D	
E	
F	

G	
H	
I	
J	
K	
L	

M	
N	
O	
P	
Q	
R	

S	
T	
U	
V	
W	
X	

Y	
Z	

Überlege dir nun eine Nachricht und ersetze („kodiere“) die Buchstaben durch die jeweilige Nummer im Alphabet. Sende die Nachricht, indem du das Programm aus Aufgabe **3** entsprechend anpasst.

Starte das Empfänger-Programm aus Aufgabe **4**. Der Empfänger muss die empfangenen Zahlen aufschreiben und „dekodieren“, also wieder durch die Buchstaben des Alphabets ersetzen.

Probiert aus, ob ihr so einen längeren Nachrichtentext korrekt empfangen könnt.