

# M5 Hacker- Fibel

# Inhalt

<b>M5Stack</b> .....	6
<b>SCHNELLSTART M5 Fire</b> .....	7

## Hardware

<b>M5 Fire</b> .....	9
<b>Display</b> .....	11
<b>SENSOR-EINHEITEN</b> .....	12
<b>SENSOR-PORT</b> .....	13
<b>GROVE KABEL</b> .....	14
<b>UNIT V2</b> .....	15
<b>LED Lichtleisten</b> .....	16
<b>LAUTSPRECHER</b> .....	17

## Software

<b>uiFlow</b> .....	19
<b>uiFlow – Einstellungen</b> .....	20
<b>uiFlow – UX Builder</b> .....	21
<b>uiFlow – Units</b> .....	22
<b>uiFlow – Befehlsbaum</b> .....	23
<b>uiFlow – Die Programmieroberfläche</b> .....	25
<b>uiFlow – Das Hauptmenü</b> .....	26

## Code

<b>Was ist Blockly?</b> .....	28
<b>Variablen</b> .....	29
<b>Schleifen</b> .....	33
<b>IF-Bedingungen</b> .....	37
<b>Ereignisse</b> .....	39
<b>Funktionen</b> .....	41

## Theorie

<b>Ada Lovelace</b> .....	44
<b>Algorithmus</b> .....	45
<b>Daten</b> .....	46

# Inhalt

<b>Künstliche Intelligenz .....</b>	<b>47</b>
<b>Bilderkennung .....</b>	<b>48</b>
<b>Großartige Anleitungen, um weiter zu lernen .....</b>	<b>49</b>

# EINLEITUNG

**Ahoi,**

Wir wissen zu wenig über die Vögel über unseren Köpfen. Ich bin Claudius Schulze. Mit meinem Team habe ich das Forschungsschiff »Zoë X« gebaut. Mit einer hochauflösenden Kamera beobachtet die "Zoë X" Zugvögel, die über den Hamburger Hafen fliegen. Dabei werden Informationen zu Orientierung, Flugwegen und der Sprache der Vögel gesammelt. Eine künstliche Intelligenz verarbeitet sie und macht sie für uns lesbar. Zum Einsatz kommt dabei eine alte Tafel, die früher an einem Flughafen die Ankunftszeiten von Flügen angezeigt hat.

Wenn du von der Hafenkante oder dem Elbstrand aus die Zoë X beobachtest, kannst du lesen, was die Vögel machen, die gerade über deinem Kopf am Himmeln vorbeiziehen.

Mit diesem Kit wollen wir dir zeigen, wie du die Technik für dein eigenes Expeditionsschiff baust und programmierst. Wir haben ein Modell der Zoë X entwickelt, das du einfach selbst zusammenbauen kannst. Es hat den Lego-Massstab 1:42. Mit dem kleinen Mikrocontroller M5 Fire kannst du Sensoren programmieren und sogar eine Kamera mit künstlicher Intelligenz trainieren, damit sie Vögel erkennt.

Entdecke mit uns die wilde Welt der Zugvögel!

**Claudius Schulze, Julia Nordholz, Jakob Vicari, Simon Boas**

Wenn du Fragen oder Ideen oder Verbesserungsvorschläge hast, schreib uns eine Email an: [info@fids-openresearchlab.org](mailto:info@fids-openresearchlab.org)

Ermöglicht wurde die Entwicklung dieses Sets durch die großzügige Unterstützung des Elbkulturfonds, Dive-In und SOMMERKINDERkultur

# Das Forschungsschiff Zoë X

Die Zoë X ist die Basis für unser FIDS Open Research Lab. Auf dem Vordeck ist unter einer Glashaube eine hochempfindliche Kamera montiert, die den Himmel filmt. Zusätzlich gibt es viele Sensoren. Sie messen zum Beispiel Lichtverschmutzung, Lärm, Temperatur, Luftdruck, die Windstärke und die Windrichtung. Und natürlich die Fahrtrichtung des Schiffes.

Der Strom kommt aus Solarzellen. Die Energie reicht nicht aus, um die Bilddaten aus der Kamera an Bord auszuwerten. Darum werden sie an einen Rechner an Land geschickt, der beim Projektprogrammierer zu Hause steht. Auf dem Dach der »Zoë X« sind dafür vier Mobilfunk.-Antennen.

Was die Kamera aufnimmt wird erstmal untersucht, ob sich überhaupt etwas verändert hat. Nur wenn das so ist, zum Beispiel, weil ein Vogel vorbeigeflogen ist, werden die Videos an Land geschickt und analysiert. Das spart Daten.

Alle Vögel werden zu Schwärmen geordnet. Kommt ein neuer Vogel ins Bild wird automatisch ermittelt ob dieser zu einem bestehenden Vogelschwarm gehört oder nicht. Dann wird der Flügelschlag untersucht. Das passiert über einen Fast Fourier Transform Algorithmus mit Sliding Window.

Aus all den Daten wird dann Text für die Anzeige gemacht. Mit Hilfe einer formalen Sprache, dem Natural Language Toolkit, werden aus verschiedenen Textbauteilen Sätze zusammengefügt. Das Ergebnis wird an das Schiff übertragen und auf der großen Ankunftstafel angezeigt.

Solche Tafeln, die üblicherweise die Ankunfts- und Abflugszeiten von Linienflügen anzeigen, heißen »Flight Information Display System«, kurz FIDS. Die Tafel auf der Zoë X hing im Flughafen Leipzig-Halle. Weil sie so alt ist, hat die Tafel ein paar Anzeigefehler (dauerhaft helle Flecken).

# M5 STACK

**M5Stack** ist ein chinesisches Technologieunternehmen. Sie stellen Minicomputer und Sensoren her und auch die Software, um sie zu programmieren. M5 Stack wurde im Jahr 2017 von Jimmy Lai gegründet und hat seinen Sitz in Shenzhen, China. Deshalb ist der rote Minicomputer M5 Fire recht billig. Der M5 Fire ist den Arduino-Computern sehr ähnlich, die dir vielleicht schon begegnet sind.

M5 heißt das Gerät, weil es 5 mal 5 Zentimeter groß ist. Stack steht für Stapeln. Du kannst nämlich verschiedene Module, zum Beispiel einen Zusatzakku oder eine Mobilfunkeinheit unter deinen M5-Computer stapeln. Dann wird daraus ein richtiger Turm. Du kannst aber auch Einheiten (englisch Units) über die Sensorports anschließen.

Viele Teile des M5 Fire sind Open Source, das heißt, du kannst den Code kostenlos nutzen, kopieren und selbst weiterentwickeln.



# SCHNELLSTART M5 Fire

Damit du den M5 Fire programmieren kannst, muss er mit dem Internet verbunden sein. Du brauchst also den Namen von deinem WLAN zu Hause oder von dort wo du mit ihm arbeitest und das dazugehörige WLAN-Passwort. Notiere dir diese Infos.

Schließe jetzt den M5 Fire mit dem USB Kabel an den Strom an, zum Beispiel, indem du ihn an den USB-Port anschliesst.. Weil der M5 Fire keine Tastatur hat, müssen wir uns eben mit dem Computer in ihn reinhacken und ihm das Passwort verraten. Und das geht so:

1. Wenn der M5 Fire startet, solltest du schnell den silbernen Knopf drücken, wenn Setup erscheint.  
**Im SETUP “ >Wifi via AP” auswählen.**
2. Verbinde deinen **Computer mit WLAN-Netzwerk M5Stack-XXXX**
3. Gebe auf deinem **Computer in irgendeinen Browser die M5 FIRE-Adresse ein: 192.168.4.1**
4. Es erscheint eine Setup-Seite. Hier wählst du dein WLAN aus, gibst dein **WLAN Passwort ein und wartest ab.**  
**Wenn die grüne Schrift erscheint:**
5. Starte den **M5 FIRE neu. Das geht mit dem roten Knopf an der Seite.**

Willkommen im Internet der Dinge! Jetzt kannst du den M5 Fire über das Internet programmieren oder ihm Befehle schicken.

## TIPP

Sollte es nicht klappen, gehe am Besten näher zum Internet-Router. Der M5 Fire ist ein kleines Gerät mit einer kleinen Antenne in der großen Weiten Welt des Internet. Der M5 FIRE beherrscht nur 2,4 Ghz Netze und mag keine Leerzeichen im Namen vom WLAN.

# Hardware



# M5 FIRE

Der M5 Fire ist ein Minicomputer mit Display, Lautsprecher, Tasten und Sensoranschlüssen. Der m5 ist modular, das heißt, du kannst andere Einheiten und Module anschliessen. Der Prozessor heißt ESP-32. Er hat leider keine Tastatur. Deshalb ist es schwieriger als auf dem Smartphone, Sachen einzugeben. Anders als andere Mikrocontroller hat der M5 FIRE ein schönes rotes Gehäuse, das halbwegs resistent ist. Durch die Löcher auf der Rückseite kannst du M5 mit Technic-Noppenbausteinen kombinieren. Und er ist sogar magnetisch! Die beiliegende Ladeplatte hält dadurch am Gerät. Der M5 Fire kann also auch am Kühlschrank oder an einer Schiffswand hängen.

Du schaltest deinen M5 Fire ein, indem du an der Seite auf den roten Knopf drückst.

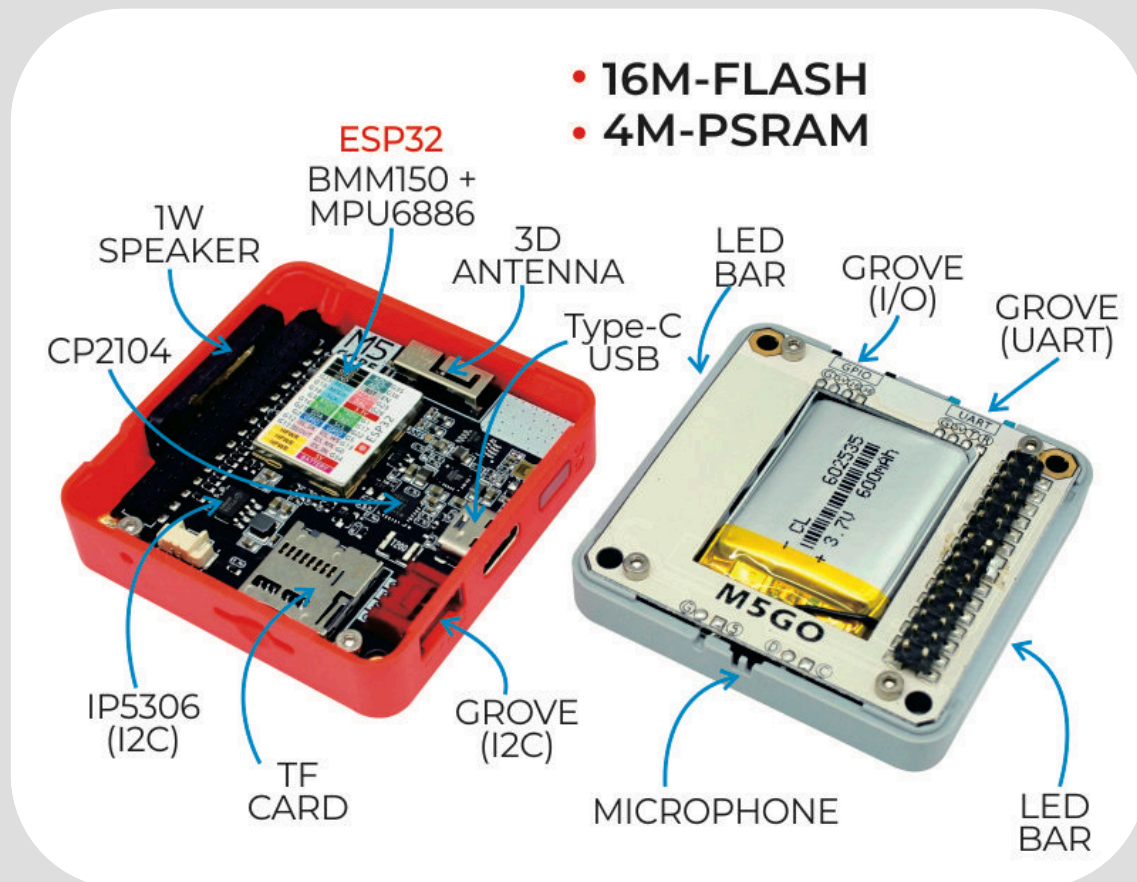
Du schaltest ihn aus, wenn du zweimal auf den Knopf drückst.

Du lädst in auf, indem du das USB-C-Kabel unten in die Ladeplatte steckst und es mit einem Ladegerät oder deinem Computer verbindest.

Du kannst deinen M5 FIRE mit dem kleinen Inbus-Schlüssel an den vier Schrauben in der Ecke aufschrauben. Dann kannst du ihn auseinanderziehen.

**Weiter auf der folgenden Seite →**

So sieht er von innen aus:



Achte beim Zusammenbauen darauf, dass der lange Stecker wieder in der langen Buchse steckt.

# DISPLAY

Der M5 Fire hat ein eingebautes 2" Display mit einer Auflösung von 320 mal 240 Pixeln. Der Punkt 0,0 ist oben links. Der Punkt oben rechts hat die Koordinaten 320,240. Das ist nicht besonders viel und kann eher ein animiertes Emoji oder ein kleines GIF abspielen, als einen Videoclip. Es reicht aber für viele Fälle aus. Vor allem ist es sehr hell. Am einfachsten ist es, Dinge mit Formen und Linien zu malen oder Text anzuzeigen. Das Display kann aber auch Bilder von der SD-Karte laden und anzeigen.

# SENSOR-EINHEITEN

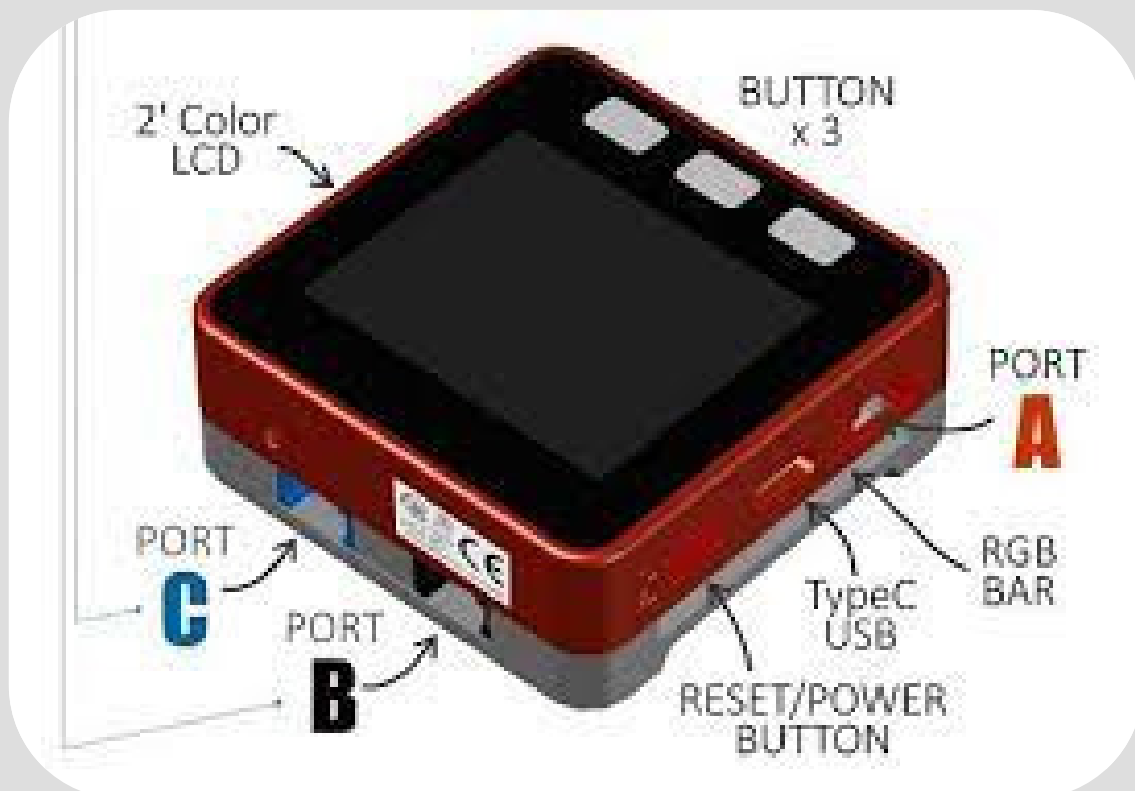
M5 stellt auch Sensor-Einheiten (englisch: Units) her. In den weißen Gehäusen stecken Sensoren, zum Beispiel für Temperatur, Farbe oder Helligkeit. Sie können mit den gleichen Grove-Kabeln an den M5 FIRE verbunden werden. Jeder Sensor spricht ein anderes Protokoll, sozusagen eine andere Sprache. Der Temperatursensor zum Beispiel ist ein Protokoll, das I2C heißt. Um einen Sensor anzuschließen:

1. Schalte den M5 Fire aus. Das geht mit zweimal Drücken auf den roten Schalter.
2. Schaue auf die Rückseite des Sensors, an welchen Port er gehört.
3. Stecke das Grove-Kabel fest in den Sensor. Dann stecke es fest in den richtigen Sensorport vom M5 FIRE.
4. Jetzt füge den Sensor im Part Units in der uifFlow-Software hinzu (siehe unten).
5. Unter den Einheiten im Befehlsbaum findest du jetzt die Blöcke in uiFlow, um den Sensor zu benutzen.



# SENSOR-PORT

Der M5 FIRE hat drei Sensoreingänge: Port A (rot) auf der linken Seite. Port B (schwarz) und C (blau) findest du auf der oberen Seite, sie sind beschriftet. Jeder Port spricht ein anderes Protokoll, eine eigene Sprache. Dazu kommt ein USB-C Eingang.



Der M5 FIRE ist merkwürdig verkabelt. Port C ist deshalb leider nicht recht benutzbar.

# GROVE KABEL

M5 Stack verwendet Grove Kabel zur Verbindung von fast allem. Sie haben vier Leitungen: Zwei für Daten, das sind die weiße und die gelbe Leitung, rot für Power, schwarz für Ground. Es gibt sie in allen Längen, von zehn Zentimetern bis zwei Metern. Grove-Kabel sind ziemlich toll: Sie sitzen schön fest an der Verbindung und können nur in einer Richtung angeschlossen werden. Merke dir: Nur weil zwei Einheiten mit dem Grove Kabel verbunden sind, heißt das nicht, dass sie miteinander funktionieren. Auf den Sensoren steht hinten drauf, in welchen Port sie gehören. Auch andere Hersteller benutzen die Kabel, zum Beispiel Seedstudio.



# UNIT V2

UNIT V2 ist eine Kamera von M5 STACK, die auf Bilderkennung mit künstlicher Intelligenz ausgerichtet ist. Sie kann Objekte, Gesichter und Bewegung erkennen, ohne mit dem Internet verbunden zu sein. Sie verwendet den Sigmstar SSD202D Prozessor. Das Betriebssystem ist Linux. Die Bilder speichert sie auf einer Micro-SD-Karte, die an der Seite hineingeschoben wird.

Sie hat oben einen Auslöser-Knopf. Sie kann mit einem Grove Kabel mit dem M5 Fire verbunden werden. Sie kann mit einem USB-C-Kabel mit einem Computer verbunden werden.



# LED Lichtleisten (RGB)

Der M5 FIRE hat 2 LED Lichtleisten. Die kannst du aufleuchten lassen. Sie funktionieren wie alle LED Lichter als Mischung aus drei Farben: rot, grün und blau. Wenn man die Farben mischt, bekommt man alle anderen Farben. Jede Farbe kann 255 Farbwerte hineingeben.

0 rot, 255 grün und 0 blau gibt einen limettengrünen Farbton. Dunkelrosa entsteht aus 255, 20, 147, Goldgelb aus 255,215,0, Kanariengelb aus 255,255,0 und Amselbraun aus 160,82,45 und Dunkelgrau aus 163,163, 163. 255 Anteile Rot, 255 Anteile Grün und 255 Anteile blau ergeben weiß. Mit schwarz (0,0,0) schaltest du die LED Leisten aus. Du kannst die eingebauten Lichtleisten mit einem Farbfeld wählen oder RGB-Werte selbst eingeben.





# LAUTSPRECHER

Kein gutes Gadget ohne Musik! Der M5 FIRE-Lautsprecher kann über die Lautsprecher-Blöcke gesteuert werden. Er gibt eher quakige Signaltöne aus, wobei du Frequenz, Dauer und Lautstärke kontrollieren kannst. Der einfachste Block ist der “Spiele Ton”-Block mit den vorgegebenen Noten, die für einen Takt abgespielt werden können. Du findest ihn unter “Hardware -> Lautsprecher”.



Spiele Ton Low A ▼ für 1 ▼ Beat

Die Vogelhochzeit geht so:

**A C A C B G B G A F C A G C C C A F F F C A G G G C B A F G E F F**

# Software

# uiFlow

uiFlow ist die Online-Programmierungsumgebung von M5 Stack. Hier kannst du Code in Blöcken zusammenschieben und auf deinen M5-Geräten laufen lassen.

Im oberen linken Bereich siehst du den UX Builder, darunter den Bereich für Units. In der Mitte ist der Befehlsbaum mit allen Befehlen. Rechts ist die Code-Oberfläche. Sie ist standardmäßig auf die Blockly-Ansicht eingestellt. Es gibt einen Tab `</>Code`, mit dem du den Code in MicroPython anschauen kannst.



Du erreichst uiFlow unter [flow.m5stack.com](https://flow.m5stack.com).

## TIPP

Du kannst deinen M5 Stack auch mit der Entwicklungsumgebung von Arduino programmieren.

# uiFlow – Einstellungen

Wenn du uiFlow aufrufst, siehst du zu Beginn das Einstellungsfenster. Hier ist es wichtig, dass du als Sprache Deutsch auswählst. Als “Gerät:” wählst du den Mikrocontroller von M5 Stack, den du vor dir hast. In diesem Set ist der rote M5 FIRE. Der API-Schlüssel ist die 8-stellige Adresse, die dir dein M5 FIRE anzeigen sollte, wenn du ihn angeschaltet hast und er erfolgreich mit dem Internet verbunden ist. Wenn nicht, blättere vor zu Grundeinstellung.


## Einstellung

[Erste-Schritte Videos](#)

API-Schlüssel : 160D574X 8/8

Sprache : Deutsch ▾

Gerät :



Abbrechen

OK

# uiFlow – UX Builder

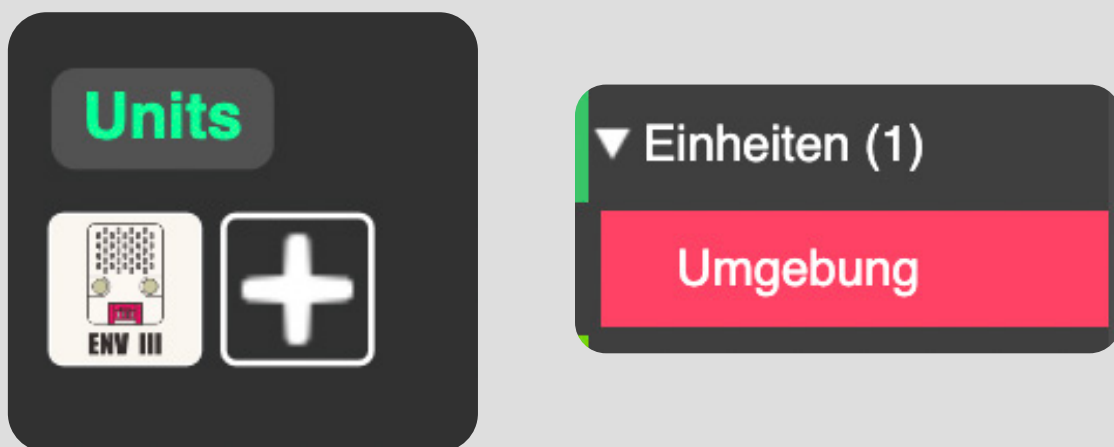
Im oberen linken Bereich siehst du den M5 Fire abgebildet. Das funktioniert wie ein Zeichenprogramm. Hier kannst du die Benutzeroberfläche für dein Programm gestalten. Du kannst aus dem kleinen Menü links Beschriftungen (Label) auf den Bildschirm ziehen. Das sind Textfelder. Was dort steht, kannst du später in deinem Programm nachträglich ändern. Wenn du auf den Text klickst, erscheint ein Popup-Fenster. Hier kannst du deinem Textfeld einen Namen geben. Bei "Text" kannst du angeben, was es anzeigen soll. Und bei der "Schrift" eine Schriftart wählen.

Du kannst aber auch Kreise (Circle), Dreiecke (Triangle) und Quadrate (Rect) auf die Fläche des Bildschirms ziehen. So kannst du einfache Zeichnungen machen. Alle Formen kannst du später im Programm verschwinden lassen oder anzeigen und ihre Farbe oder ihre Position ändern.



# uiFlow – Units

Der Bereich Units befindet sich links unten in uiFlow. Hier kannst du Sensoren, Lichter, Motor und Taster an deinen M5 FIRE anschließen. Drücke auf das +-Zeichen und ein Fenster erscheint. In diesem sind alle Einheiten gezeigt, die du an den M5 Fire anschließen kannst, Finde deine Einheit, zum Beispiel den ENVIII-Sensor und wähle ihn aus, indem du in das weiße Feld klickst bis ein grüner Haken erscheint. Der Name der Einheit steht meist auf der Rückseite.



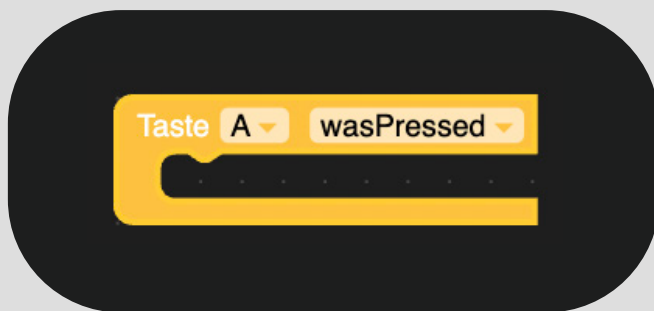
Der ENVIII Sensor kann drei verschiedene Werte ausgeben. Diese stehen als Code-Puzzlestücke zur Programmierung zur Verfügung: Druck, Temperatur und Feuchtigkeit.



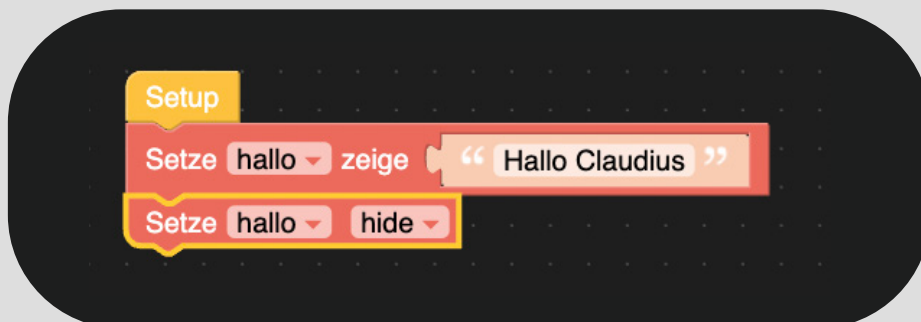
# uiFlow – Befehlsbaum

Der Befehlsbaum in der Mitte enthält die Befehle, die du in deinem Programm benutzen kannst als fertige Blöcke. Du kannst sie nehmen und auf die Programmieroberfläche schieben.

Ob eine der drei silbernen Tasten unter dem Display gedrückt wurde, kannst du unter “Ereignis” nutzen. Taste A ist die ganz links, Taste B ist die mittlere und Taste C ist ganz rechts.

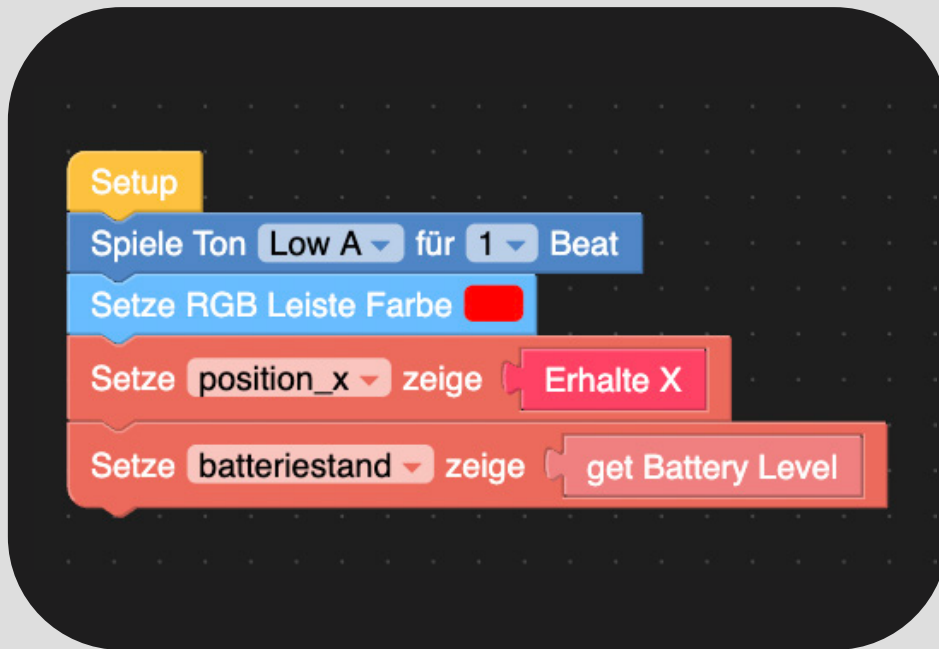


Unter “UI” (das ist englisch und steht für User Interface, also Benutzeroberfläche) findest du alle Elemente, die du im UX-Builder gezeichnet als Codeblöcke hast. Hier kannst du zum Beispiel den Textfeldern einen neuen Text geben oder sie verschwinden lassen.



**Weiter auf der folgenden Seite →**

Unter “Hardwares” findest du alle Funktionen, die dein M5 FIRE kann. So kann er Töne erzeugen (Lautsprecher), die LED-Lichteisten an den Seiten anschalten (RGB) und den Batteriestand abfragen (Power). Er hat auch einen Lagesensor, der angibt, in welcher Position der M5 Fire gerade liegt und sogar, ob er gerade bewegt wird (IMU).



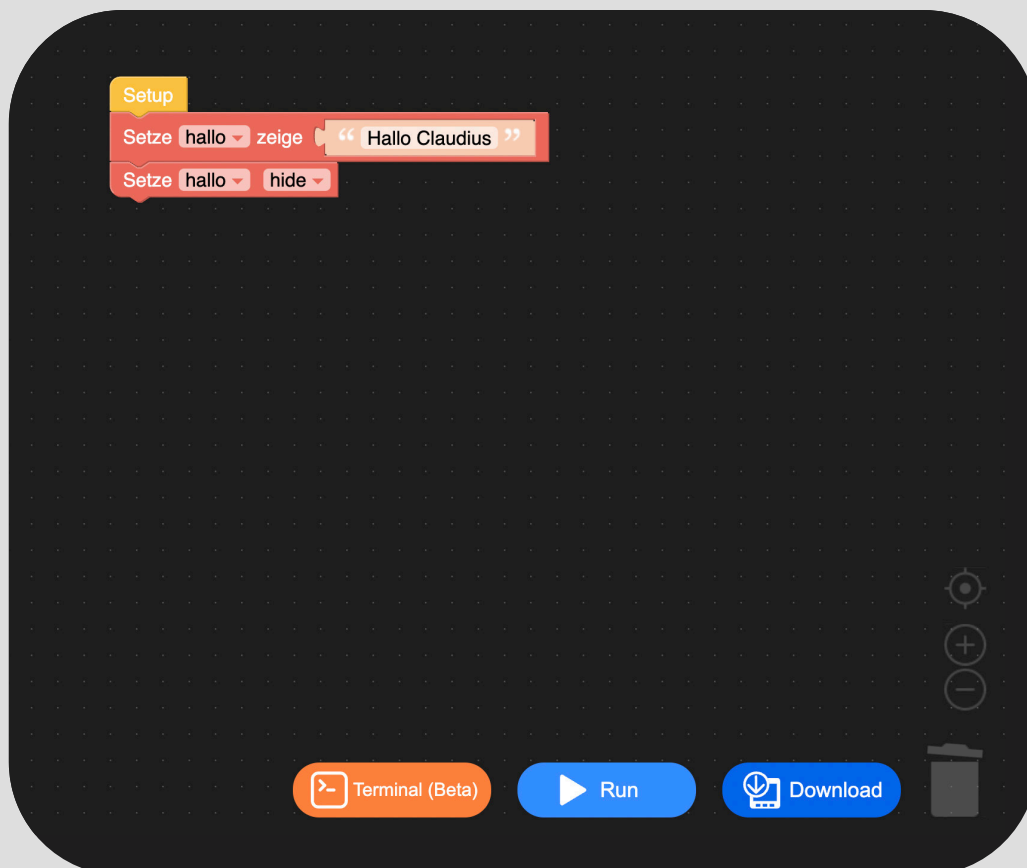


# uiFlow – Die Programmieroberfläche

Auf die Programmieroberfläche kannst du die Code-Blöcke aus dem Befehlsbaum schieben. Wenn du sie an den gelben Block “Setup” andockst, werden sie nacheinander ausgeführt.

Setup ist der erste Befehl, “Setze hallo zeige ‘Hallo Claudius’” ist der zweite Befehl. Setze “hallo hide” ist der dritte Befehl.

Wenn du auf die blaue Schaltfläche “> Run” drückst, wird dein Programm auf den M5 Fire übertragen und dort einmal live ausgeführt. Wenn du auf den Button “Download” drückst, wird dein Programm auf den M5 FIRE geladen, dort gespeichert und immer wieder ausgeführt, wenn du ihn anschaltest. Du kannst Blöcke in die graue Papiertonne unten rechts schieben, um sie zu löschen.



# uiFlow – Das Hauptmenü

Das Hauptmenü von uiFlow ist verwirrend. Es sind viele Funktionen enthalten. Du brauchst allerdings erst einmal nur wenige. Ganz links steht die aktuelle Versionsnummer von uiFlow. Neben "Projekt" findest du ein Feld, in das du den Namen deines Programms schreiben solltest. Daneben ist der Tab, der zwischen der "Blockly"-Ansicht und der Code-Ansicht in "</> Python" hin und her schaltet. Ein Klick auf das Floppy-Disk-Symbol speichert dein Programm auf der Festplatte. Ein Klick auf den Ordner öffnet eine Programmdatei von deinem Rechner. Das Symbol Doc führt dich zur englischsprachigen Dokumentation. Ganz rechts, im Burger-Menü mit den drei Strichen kommst du zum uiFlow Einstellungsfenster.



# Code

# Was ist Blockly?

Wenn du Programmieren hörst, denkst du sicher an lange Codezeilen. Blockly verpackt die langen Codestücke in Blöcke. Diese Blöcke kannst du per Drag-and-Drop zusammensetzen. So kannst du viel schneller schwierige Dinge programmieren. Wenn du ein Rechteck in uiFlow anzeigen willst, schiebst du es im UX-Builder an die richtige Stelle. Als Code müsstest du

**rectangle0 = M5Rect(243, 34, 30, 30, 0xFFFFFF, 0xFFFFFF)**

schreiben. Wenn du die Größe des Rechtecks auf Breite 50, Länge 50 verändern willst, kannst du diesen Block benutzen:



Im Code müsstest du stattdessen Folgendes schreiben – und zwar mit richtiger Groß- und Kleinschreibung:

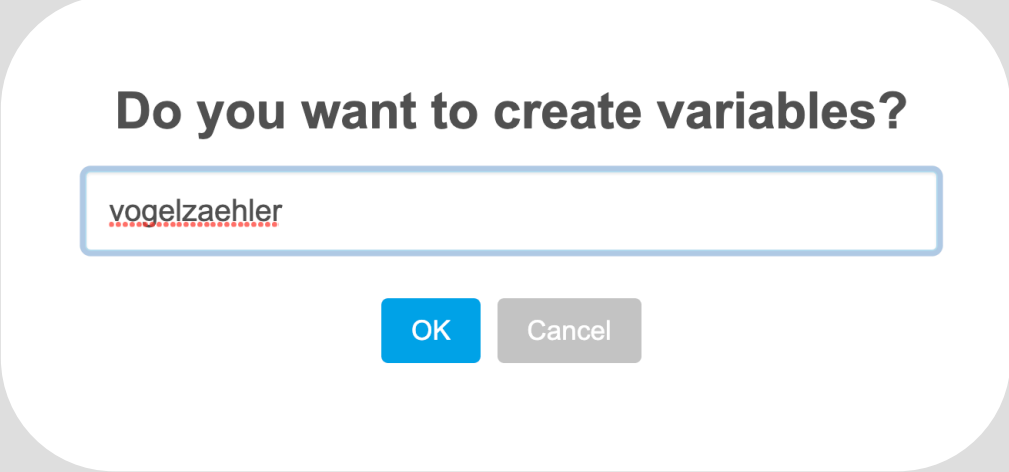
**rectangle0.setSize(50, 50)**

Die Blöcke sind also Verpackungen für Code. Manchmal sogar für mehrere Zeilen. Sie lassen sich schneller schieben und besser lesen. Das Programmieren mit Blockly ist also auch richtiges Programmieren! uiFlow verpackt den Code der Programmiersprache Micropython in die Blöcke.

# Variablen

Variablen sind die Kästen für deine Daten. Immer wenn du etwas misst, legst du dafür Variablen fest, die Werte speichern.

So machst du eine Variable: Du gehst im Befehlsbaum auf “Variablen” und klickst “Erzeuge Variable...”. Dann geht ein Fenster auf, und du gibst der Variablen einen Namen. Zum Beispiel “vogelzaehler”. Dann klickst du auf “OK”. Dieser Name steht dann draußen am Kasten. Namen von Variablen dürfen keine symbolischen Zeichen, kein äöüß und keine Leerzeichen enthalten.

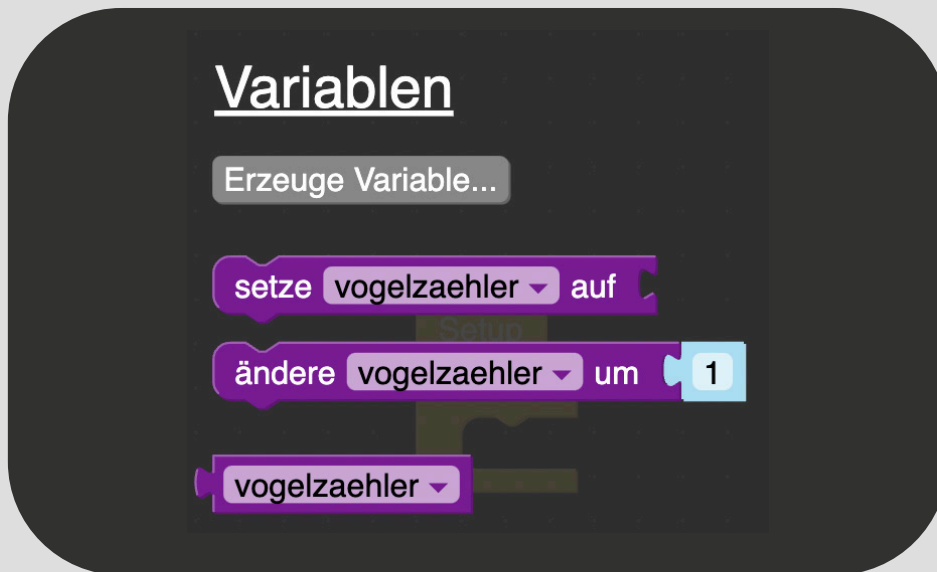


A screenshot of a software dialog box. The title bar reads "Do you want to create variables?". Below the title is a text input field containing the text "vogelzaehler". At the bottom of the dialog are two buttons: "OK" (highlighted in blue) and "Cancel" (greyed out).

**Weiter auf der folgenden Seite →**

Jetzt bekommst du die Blöcke mit deiner Variablen “vogelzaehler” im uiFlow-Menü angezeigt. Du kannst den Wert setzen mit dem Block “setze vogelzaehler auf”. Dafür brauchst du noch ein Puzzlestück mit einer Zahl. Das findest du im Befehlsbaum bei “Mathe”.

Du hast einen fertigen Block, um den Wert um 1 zu erhöhen: “ändere vogelzaehler um 1”. Und du hast das Puzzleteil “vogelzaehler”, um die Variable irgendwo zu benutzen.



**Weiter auf der folgenden Seite →**

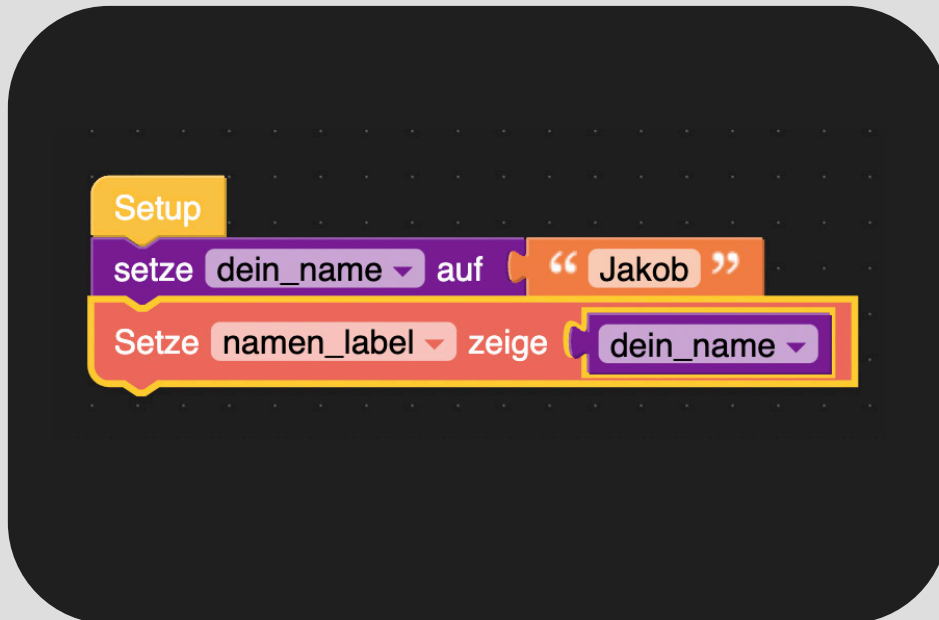
Den Inhalt deiner Variablen kannst du auf dem Bildschirm anzeigen. Dieses Programm setzt am Anfang einmal den Wert der Variablen “vogelzaehler” auf 0. Dann beginnt die Schleife. Die Variable wird um den Wert 1 erhöht und angezeigt. Dann wartet das Programm eine Sekunde. Dann wird die Variable um 1 erhöht und angezeigt. Dann wartet das Programm eine Sekunde. Dann wird die Variable um 1 erhöht und angezeigt. Dann wartet das Programm eine Sekunde. Dann wird die Variable um 1 erhöht und angezeigt. Dann wartet das Programm eine Sekunde. Dann wird die Variable um 1 erhöht und angezeigt.

Ich glaube, du hast das Prinzip verstanden.



**Weiter auf der folgenden Seite →**

Variablen können natürlich auch Text enthalten, zum Beispiel einen Namen. Somit ist es dem Programm möglich, dich oder andere Personen mit richtigem Namen zu begrüßen. Das sieht dann so aus:





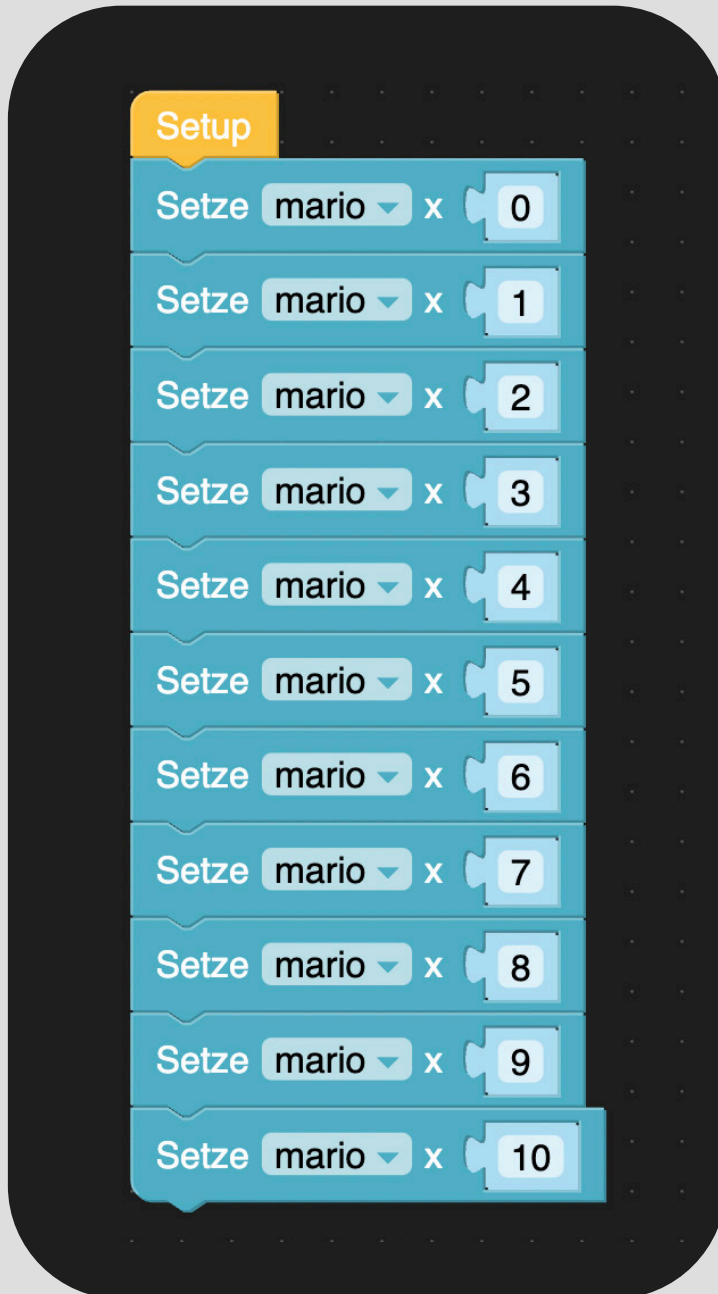
# Schleifen

Schleifen brauchst du, wenn du einen Codeblock öfter als einmal ausführen möchtest. Stell dir vor, du möchtest den roten Kreis mit dem Namen “Mario” Schritt für Schritt über den Bildschirm bewegen lassen.



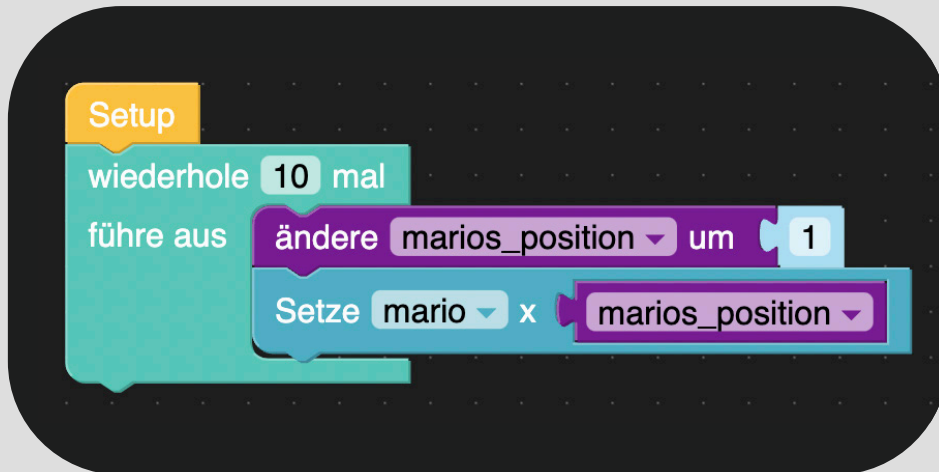
**Weiter auf der folgenden Seite →**

Dann könntest du die x-Position mit einem Befehl jedes Mal rändern. Das sieht dann so aus: Oder du benutzt eine Schleife.



**Weiter auf der folgenden Seite →**

Eine Schleife wiederholt einen Codeblock entweder eine bestimmte Anzahl lang oder bis eine Bedingung eintritt. Wenn du Mario zehn Schritte nach rechts bewegen willst, sieht das so aus: Das typische Programm fängt beim Setup-



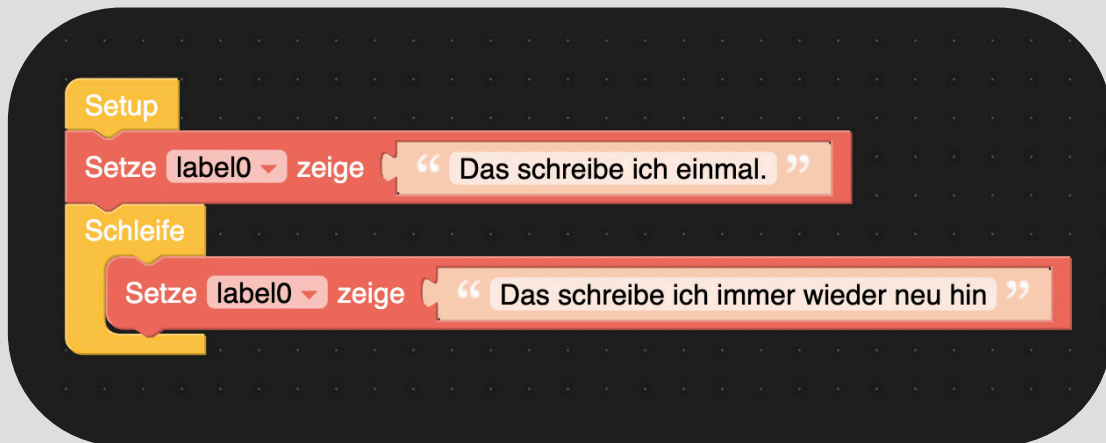
Soll es zehnmal wiederholt werden: Zuerst wird Marios Position um 1 erhöht, dann wird der Kreis Mario auf die neue Position gesetzt.

Wenn du dein Programm länger laufen lassen möchtest, schickst du es in eine Dauerschleife. Sie heißt in uiFlow einfach "Schleife".



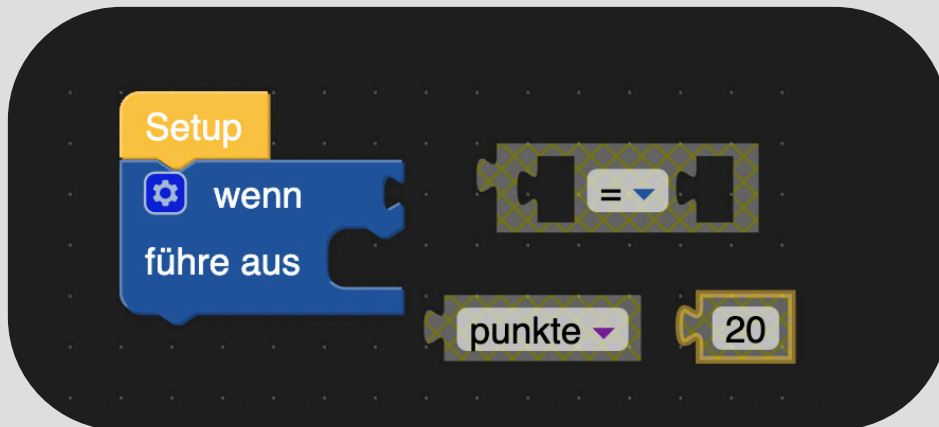
**Weiter auf der folgenden Seite →**

Block an. Danach kannst du Blöcke anhängen, die genau einmal, nämlich zu Beginn, ausgeführt werden. Dann folgt die gelbe Schleife. In sie legst du alles hinein, was regelmäßig ausgeführt werden soll. Zum Beispiel die Funktion “Nachschauen, ob ein Vogel aufgetaucht ist.”



# IF-Bedingungen

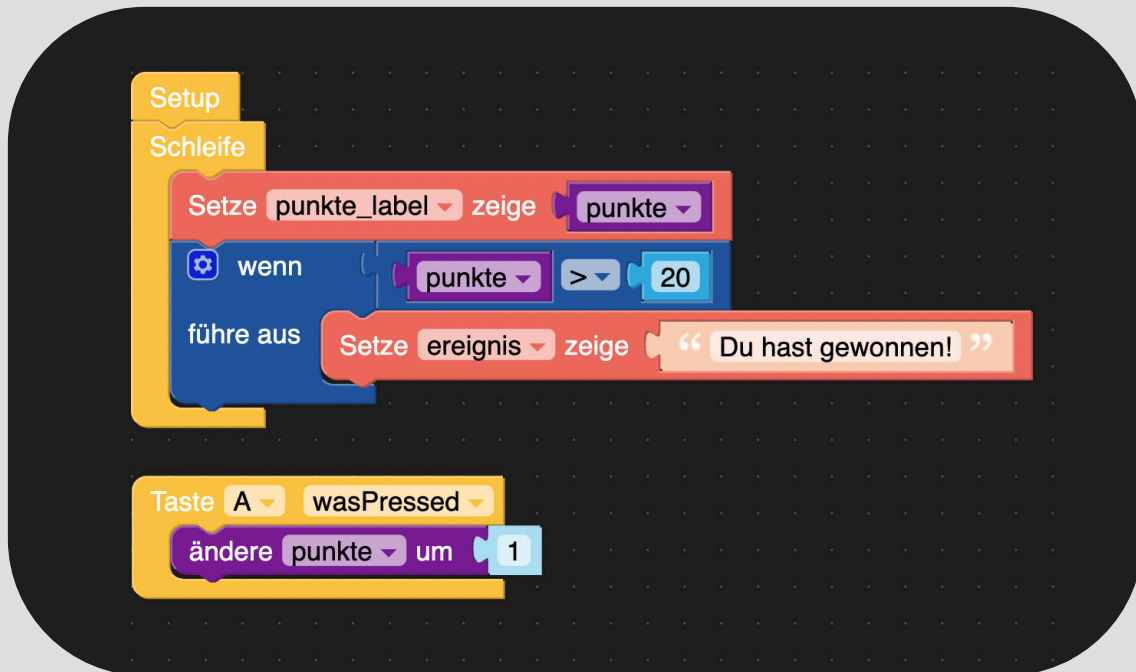
Oft musst du selbst prüfen, ob ein Ereignis eintritt. Zum Beispiel: ... Wenn du die Taste drückst, soll das Licht angehen. ... Wenn du 20 Punkte hast, hast du das Spiel gewonnen. ... Wenn du einen gelben Vogel siehst, soll die Variable "gelbe\_voegel" um 1 erhöht werden. Dafür gibt es den Super Logik-Block "wenn das, führe aus dies".



**Weiter auf der folgenden Seite →**

Daraus wird das einfachste Spiel der Welt: Du musst auf eine Taste an deiner Spielekonsole, dem M5 FIRE, drücken. Du hast gewonnen, wenn du zwanzig Mal die Taste drückst. Klingt einfach? Die Zubehör-Puzzle-Blöcke findest du in Variablen, Mathe und Logik.

Es wird so programmiert:



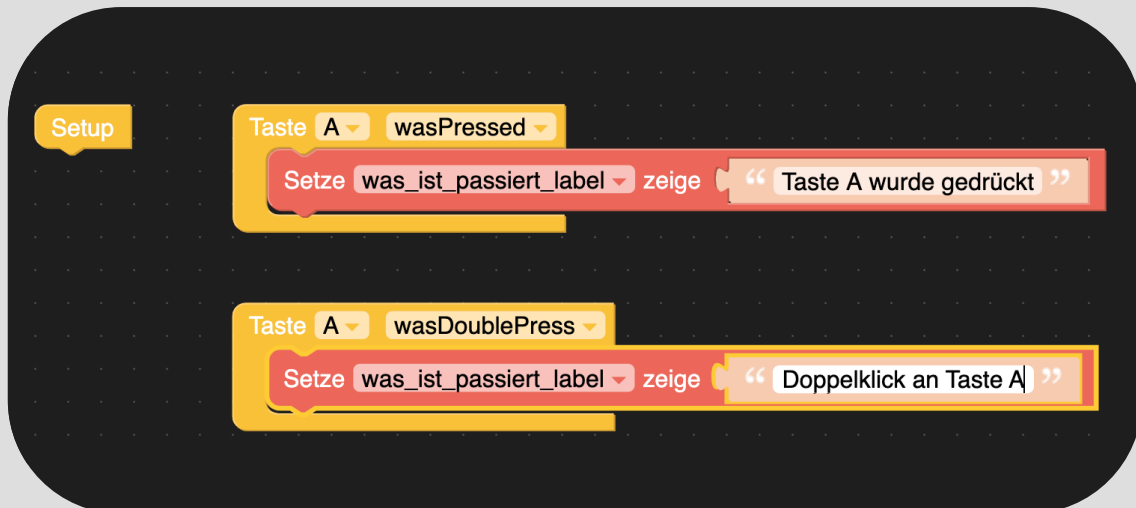
Der Blaue Logik-Block prüft, ob die Zahl der Punkte 20 überschreitet. Dann, beim 21. Tastenklick, wird dir angezeigt: "Du hast gewonnen." Besiegst du den Endgegner?

# Ereignisse

Ereignisse treten ein, wenn zum Beispiel jemand eine Taste drückt. oder wenn eine bestimmte Zeit verstrichen ist. Ereignisse finden sich in Blockly unabhängig vom normalen Programm-Strang in den gelben Ereignisklammern. Wenn Taste A am M5 Fire gedrückt wird (das ist die silberne Taste ganz links), soll er Folgendes anzeigen.

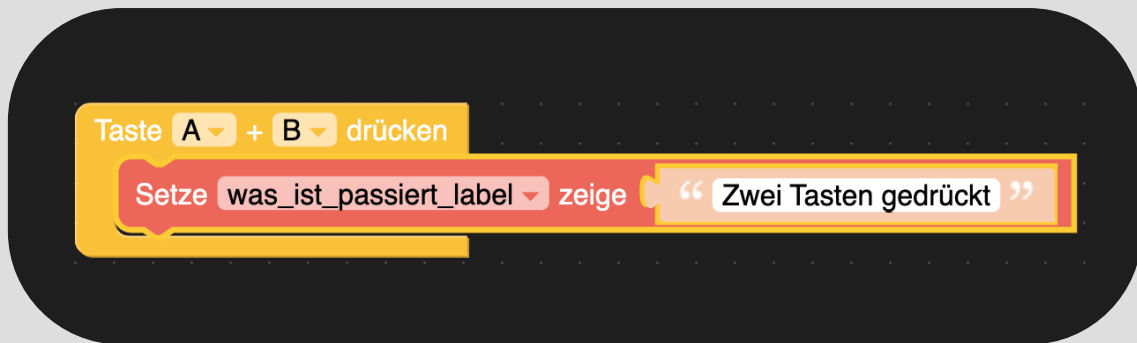


Damit du mit drei Tasten möglichst viel machen kannst, kannst du auch reagieren, wenn die Taste zweimal gedrückt wird oder losgelassen wird.



**Weiter auf der folgenden Seite →**

Du kannst sogar reagieren, wenn zwei Tasten gedrückt werden.



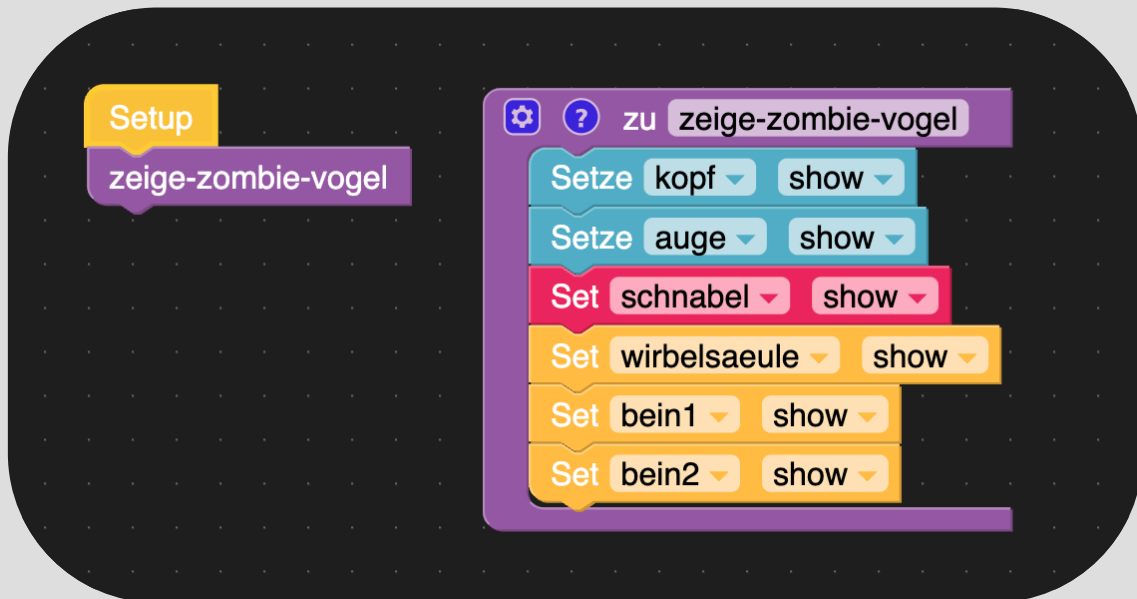


# Funktionen

Bisher lief dein Programmcode einfach von oben nach unten, sodass Block nach Block ausgeführt wurden ausgeführt. Es kann auch der Fall eintreten, dass du manche Sachen öfter ausführen willst. Dann kannst du entweder alle Blöcke an die neuen Stelle kopieren Oder du packst sie in eine Funktion. Das ist quasi ein kleines Programm in deinem größeren Programm, das du aufrufst, um etwas bestimmtes zu erledigen, wie zum Beispiel den Befehl “den Zombie-Vogel anzeigen, wenn dein Spiel vorbei ist”.



**Weiter auf der folgenden Seite →**



In der Funktion „zeige-zombie-vogel“ sind alle Befehle gesammelt, die die Elemente des Vogels anzeigen. Bisher besteht der Vogel nur aus sechs Elementen. Stell dir vor, er bestünde aus 20 Elementen. Eine zweite Funktion könnte heißen „verstecke-zombie-vogel“. Sie blendet ihn aus. Und zwar so oft du willst.

Ein weiterer Vorteil von Funktionen: wenn du deinen Vogel verbesserst, kannst du das an einem Ort tun. Und musst nicht überall, wo der Vogel gemalt wird, Veränderungen vornehmen.



# Theorie

# Ada Lovelace

Ada war eine der ersten Programmiererinnen. Sie wurde 1815 in London geboren und heißt eigentlich Augusta Ada King-Noel, Countess of Lovelace. Zu Adas Zeiten durften Frauen in England noch nicht studieren. Adas Mutter brachte ihr aber eine Menge bei. Als sie 13 Jahren alt war, erfand sie eine Art Drone, eine dampfgetriebene Flugmaschine. Und sie schrieb die „Wissenschaft“ der Flugologie. Später arbeitete sie mit Programmen, die als Löcher in Karten gestanzt waren. So wurden automatische Webstühle gesteuert. Ada Lovelace machte sich schon vor fast 200 Jahren Gedanken über die Intelligenz von Maschinen. Sie schrieb: „Die Maschine kann das tun, was wir ihr zu befehlen vermögen, sie kann der Analyse folgen. Sie hat jedoch keine Fähigkeit zur Erkenntnis analytischer Verhältnisse oder Wahrheiten.“ Sie meint, dass Maschinen ganz anders sind als Menschen. Weil sie kein Bauchgefühl, Intuition, haben, können sie nichts selbst herausfinden. Wichtige Gedanken auch für unser Forschungsschiff. Was denkst du darüber?

# Algorithmus

Ein Algorithmus ist eine Schritt-für-Schritt Folge, um ein Problem zu lösen. Wenn du die Entfernung eines Vogels zum Vogelhaus anzeigen willst, brauchst du dafür einen Algorithmus. Du findest ihn in der Anleitung zum Ultraschall-Sensor. Es gibt auch komplexere Algorithmen. Sie sind zum Beispiel am Werk, wenn die V2-Kamera ein Objekt erkennt. Ein Algorithmus besteht aus verschiedenen Operationen. Als Beispiel ein Algorithmus, mit dem eine Roboterhand Saft einschenken soll: "Öffne eine Flasche Saft" ist eine Operation. "Bewege die Flasche in Richtung von Claudius" ist eine zweite Operation. "Giesse den Saft in das Glas von Claudius" eine dritte Operation.

# Daten

Alles was du mit deinem Gadget sammelst sind Daten: Messwerte, Bilder, Sounds. Daten sind die Grundlage für die Wissenschaft. Die Daten enthalten Informationen über die echte Welt. Du musst sie interpretieren, also entschlüsseln. Wenn dein Farbsensor am Forschungsschiff rot: 165, grün: 42, blau: 42 meldet, kannst du interpretieren, dass das Amselbraun ist. Eine Amsel ist also wahrscheinlich gelandet. Das Problem mit der Interpretation: Es könnte auch ein anderer amselbrauner Vogel oder eine braune Papiertüte sein.

# Künstliche Intelligenz

Die Künstliche Intelligenz gehört zur Computerwissenschaft. Sie entwickelt Technologie, die Aufgaben so intelligent lösen soll, wie ein Mensch. Künstliche Intelligenz steckt heute unsichtbar in vielen Dingen: in der Smartwatch und im Online-Shop. Zur Intelligenz gehören: überlegen, planen, Rätsel lösen und erkennen. Damit die Künstliche Intelligenz das kann, müssen Menschen sie programmieren. Und, wie in unserem Fall.

# Bilderkennung

Die Bilderkennung ist eine Art der Künstlichen Intelligenz. Damit Bilderkennung klappt, braucht man eine Kamera. Und ein Modell, nach dem die Kamera Objekte und Vögel sucht.

Erst brauchst du Material zum Training deines Modells: Möglichst viele Fotos von allen Seiten des Objekts. Dann musst du die Bilder annotieren. Du musst dabei der Kamera sagen, was auf den Bildern zu sehen ist: Eine Krähe, eine Taube, zwei Küken. Daraus errechnet ein großer Computer dann ein Modell. Dieses Modell ist eine Gehirn-Datei. Du kannst sie auf deine V2-Kamera laden. Und schon ist sie intelligent. Du kannst das Modell auch auf Kameras von deinen Freunden laufen lassen. Stell dir vor, es gäbe solche Modelle für Menschen. Dann könntest du eine Sprache lernen einfach per Upload in dein Gehirn.



# Großartige Anleitungen, um weiter zu lernen

Jakobs Blog übers Geschichtenerzählen mit Sensoren:

<https://www.riffreporter.de/de/magazine/journalismus-digitalisierung>

Die M5-Stack Anleitung (auf Englisch):

<https://docs.m5stack.com/>

M5-Stack-Projekte auf hackster.io:

<https://m5stack.hackster.io/projects>

Die Hardware von M5 Stack könnt ihr z.B. bei Eckstein-Komponente kaufen:

<https://eckstein-shop.de/>

Lustige Anleitungen (auf Englisch) hat Adafruit:

<https://learn.adafruit.com/>

Literaturtipps

Cosicosa & Ana Seixas. Hallo Roboter. Schlaue Maschinen und Künstliche Intelligenz. Kleine Gestalten.

## TIPP

Literaturtipps:

*Cosicosa & Ana Seixas. Hallo Roboter. Schlaue Maschinen und Künstliche Intelligenz. Kleine Gestalten.*

# Impressum

**Claudius Schulze**

Künstler

**Jonathan Kossick**

IT, FIDS Softwareentwicklung &  
Maschinelles Lernen

**Sithara Pathirana**

Projektleitung

**Dr. Jakob Vicari**

Entwicklung & Prototyping

**Simon Dirks**

M5 Softwareentwicklung

**Julia Nordholz**

pädagogische Begleitung

**Lisa Beese**

Projektkoordination & Controlling

**Kummer & Herrman**

Grafikdesign

**fids-openresearchlab.org**

## Bildrechte

Alle Bilder von unsplash.com  
außer wenn anders angegeben.  
Das Lernpaket macht Gebrauch  
von einer CC BY-NC-SA 3.0 DE  
Lizenz.



