

Übersicht der Programmierung des TXT 4.0 mit ROBO Pro Coding (Kurzreferenz)

Zusammengestellt von A. Chobe
(axel.chobe@t-online.de / chobe.info)

Aktuelle Version vom 01.03.2024



Inhalt

1	Oberfläche	6
1.1	Menü.....	6
1.1.1	Voreinstellungen/Einstellungen	6
1.1.2	Projekte	7
1.1.3	Hilfe	7
1.2	Blaue Leiste	8
1.2.1	Programm starten.....	8
1.2.2	Programm beenden	8
1.2.3	Debugging.....	8
1.2.4	Schnittstellentest	8
1.2.5	Programm hochladen)	8
1.2.6	Controller verbinden	8
1.3	Rechte Seite	8
1.4	Hauptprogramm	9
1.4.1	Lernstufen	9
1.4.2	Drucken	9
1.4.3	Schritt zurück/vor	9
1.4.4	Zentrieren.....	9
1.4.5	Verkleinern/ Vergrößern	9
1.5	Controllerkonfiguration	9
1.6	Kamerakonfiguration.....	10
1.6.1	Farberkennung.....	10
1.6.2	Bewegungserkennung.....	10
1.6.3	Ballerkennung.....	10
1.6.4	Linienerkennung	10
1.6.5	Aussparung.....	10
1.7	Anzeigeconfiguration.....	11
1.7.1	Inspektor	11
1.7.2	Button	11
1.7.3	Schiebeschalter	11
1.7.4	Kontrollfeld.....	11
1.7.5	Schieberegler	11
1.7.6	Label	11
1.7.7	Input.....	11
1.7.8	Status Indikator.....	11
1.7.9	Messinstrument (Gauge).....	11
1.7.10	Bilddarstellung.....	11
1.8	Bedienfeld	12
1.9	Konsole.....	12
1.10	Haltepunkte.....	12
1.11	Ausdrucke	12
1.12	Aufrufstapel.....	12

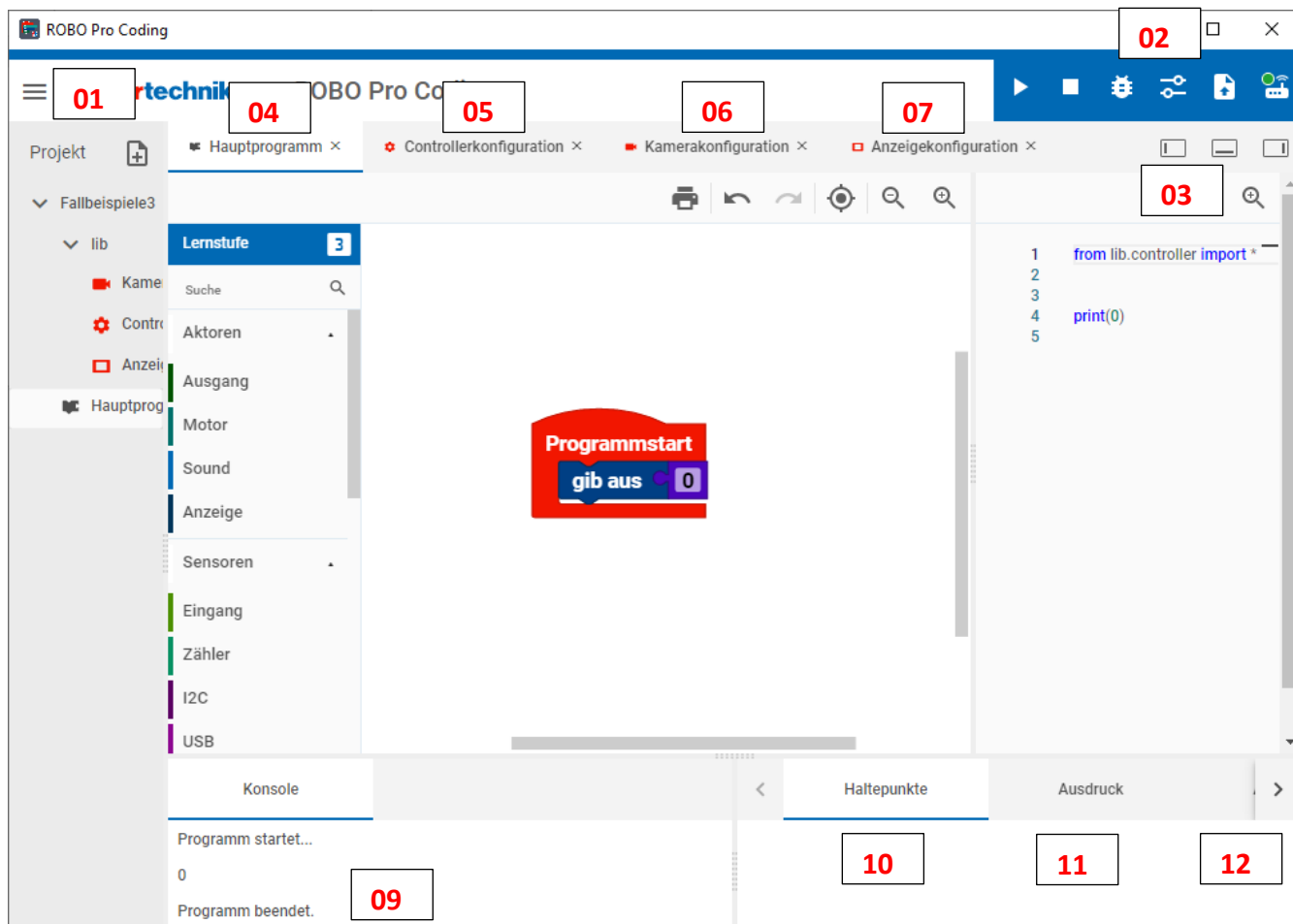
2	Informationen zum Umgang mit Blöcken.....	13
2.1	Arten der Blöcke	13
2.2	Block vervielfältigen.....	13
2.3	Kommentar hinzufügen	13
2.4	Veränderung des Aussehens.....	13
2.5	Verkleinerte Darstellung	13
2.6	Baustein löschen.....	13
3	Aktoren	14
3.1	Ausgang.....	14
3.1.1	LED	14
3.1.2	Motor	14
3.1.3	Magnetventil	14
3.1.4	Kompressor.....	14
3.2	Motor	14
3.2.1	Motor (Bidirektional)	14
3.2.2	Encoder Motor.....	14
3.2.3	Servomotor.....	15
3.3	Sound	15
3.4	Anzeige.....	17
3.4.1	Button (txt_button).....	17
3.4.2	Schalter (txt_switch)	17
3.4.3	Kontrollfeld (txt_checkbox)	17
3.4.4	Schieberegler (txt_slider)	18
3.4.5	Beschriftungsfeld (txt_label)	18
3.4.6	Eingabefeld (txt_input)	18
3.4.7	Statusanzeige (txt_status_indicator)	18
3.4.8	Messinstrument (txt_gauge)	18
4	Sensoren	19
4.1	Eingang.....	19
4.1.1	Taster	19
4.1.2	Ultraschallsensor	19
4.1.3	Farbsensor	19
4.1.4	Spurensucher.....	19
4.1.5	Fototransistor	20
4.1.6	Fotowiderstand.....	20
4.1.7	NTC Widerstand.....	20
4.2	Zähler	21
4.2.1	Hole Zählerwert	21
4.2.2	Ist Zähler ... Wert.....	21
4.2.3	Setze Zähler zurück	21
4.3	I ² C.....	22
4.3.1	Kombisensor	23
4.3.2	Umweltsensor.....	23
4.3.3	Gestensensor	23

4.4	USB (Kamera)	24
4.4.1	Bewegungserkennung	24
4.4.2	Farberkennung	24
4.4.3	Ballerkennung	24
4.4.4	Linienerkennung	24
4.4.5	Bilddarstellung auf Controller	25
4.4.6	Mikrofon	25
5	Verarbeitung	26
5.1	Logik	26
5.1.1	Boolesche Logik	26
5.1.2	Wenn Funktion	26
5.1.3	Vergleichsoperatoren	26
5.1.4	Logische Operatoren	26
5.2	Schleifen	26
5.2.1	Dauerhafte Wiederholung (Dauerschleife)	26
5.2.2	Wiederhole (Zählschleife)	26
5.2.3	Wiederhole solange (Kopfschleife)	26
5.2.4	Wiederhole bis (Fußschleife)	26
5.2.5	Zählschleife von bis (For-Schleife)	26
5.3	Mathe	27
5.3.1	Einfache Rechnungen	27
5.3.2	Spezielle Rechnungen	27
5.3.3	Operatoren *1	27
5.3.4	Auswertung von Listen *2	27
5.3.5	Funktionen	27
5.3.6	Konstanten	27
5.3.7	Konvertierung	27
5.3.8	Zahl prüfen	27
5.3.9	Zufallszahlen	28
5.3.10	Rest einer Division	28
5.3.11	Zahl runden	28
5.3.12	Werte beschränken	28
5.3.13	Wertebereich wandeln	28
5.3.14	Atan2	28
5.4	Text	29
5.4.1	Textausgabe	29
5.4.2	Textlänge ermitteln	29
5.4.3	Prüfen auf leeren Text	29
5.4.4	Suche im Text	29
5.4.5	Extrahieren einzelner Zeichen	29
5.4.6	Extrahieren eines Textteils	29
5.4.7	Groß und Kleinschreibung ändern	30
5.4.8	Leerzeichen entfernen (vorn bzw. hinten)	30
5.4.9	Schriftgröße ändern	30
5.4.10	Texte formatieren	30

5.5	Datei.....	31
5.5.1	Datei ... mit Modus... öffnen	31
5.5.2	schreibe... in Datei... neue Zeile am Ende.....	31
5.5.3	Datei ... schließen.....	31
5.5.4	Datei... als Text lesen.....	32
5.5.5	Liste... zeilenweise in die Datei ... schreiben.....	32
5.5.6	Datei ... zeilenweise als Liste lesen	32
5.5.7	Verzeichnis path erstellen	33
5.5.8	existiert	33
5.5.9	Beispiel zur Datenaufnahme	33
5.6	Datenstrukturen (Listen)	34
5.6.1	Liste erzeugen.....	34
5.6.2	Länge einer Liste ermitteln.....	34
5.6.3	Suche und manipulieren von Elementen in einer Liste.....	34
5.6.4	Sublist erstellen	35
5.6.5	Liste sortieren	35
5.6.6	Liste drucken.....	35
5.7	Util	36
5.7.1	Farbauswahl	36
5.7.2	Warte (Zeit]	36
5.7.3	Warte (Bedingung).....	36
5.7.4	Thread	36
5.8	Variablen.....	36
5.8.1	Erstellen.....	36
5.8.2	Festlegen	36
5.8.3	Wert ändern	36
5.9	Funktionen (Unterprogramme).....	37
5.9.1	Einfacher Aufruf.....	37
5.9.2	Aufruf mit Rückgabe	37
5.9.3	Aufruf mit Rückgabe und Bedingung	37
5.9.4	Funktion als Thread ausführen	37
5.10	Machine Learning	37
5.11	Importe	38
6	Kommunikation.....	39
6.1	Fernbedienung	39
6.1.1	remote_button	39
6.1.2	remote_slider	39
6.1.3	remote_label	39
6.1.4	remote_status_indicator.....	40
6.1.5	remote_image	40
6.1.6	remote_joystick.....	40
6.1.7	remote_chart.....	41
6.2	Sprachsteuerung.....	41
6.3	Cloud / MQTT	42

Außer der Bedienungsanleitung (<https://www.fischertechnik.de/de-de/service/elearning/lehren/txt-40-controller>) des Controllers und einer Einleitung zu ROBO Pro Coding (<https://www.fischertechnik.de/de-de/service/elearning/lehren/base-set-and-add-ons>) gibt es noch keine Anleitung, um ROBO Pro Coding zu erlernen. Auch die vorgefertigten Programme zu den Modellen sind nicht immer hilfreich, um die Grundlagen zu verstehen. Mit dem folgenden Heft habe ich versucht ein einfaches und umfangreiches Dokument zu erstellen, um ein besseres Verständnis für die Programmiersprache zu erhalten.

1 Oberfläche



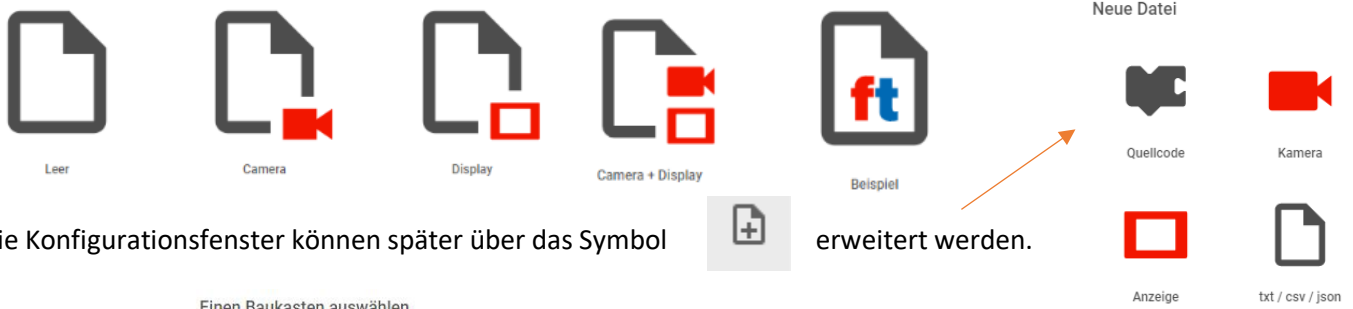
1.1 Menü (01)

1.1.1 Voreinstellungen/Einstellungen



1.1.2 Projekte

- **Neu** (Leer; Camera; Display; Camera + Display; Beispiel)



Die Konfigurationsfenster können später über das Symbol



erweitert werden.

Einen Baukasten auswählen

STEM Robotics Add On Autonomous Driving



Unter dem Punkt Beispiele können verschiedene Programme für die entsprechenden Baukästen aufgerufen werden. Zuerst den Kasten wählen und dann das entsprechende Programm.

Servo_Calibration



ABBRECHEN

VORHERIGE

NÄCHSTE



Quellcode

anzulegen.

Um den Quellcode übersichtlicher zu gestalten, ist es möglich zusätzliche Module hier können Funktionen geschrieben werden, die dann im Hauptprogramm aufgerufen werden. (Siehe Punkt Import) Alle Projektteile (Hauptprogramm, Controllerkonfiguration usw.) werden in einer Libery gespeichert.

- **Exporte** (Lokal; fischertechnik GitLab)



Lokal



fischertechnik GitLab

Die erstellten Programme, können auf dem eigenen PC oder unter fischertechnik GitLab abgespeichert werden. Für diesen Fall muss man sich dort anmelden. Dazu kann man sein Fischertechnikzugang auswählen.



- **Importe** (Lokal; fischertechnik GitLab)

Auch das speichern der Programme erfolgt entweder lokal oder auf GitLab.

- **Schließen** (Projekt schließen)

Vorher erfolgt die Abfrage, ob das Projekt gespeichert werden soll.

Bist du sicher, dass du das Projekt schließen möchtest?

Deine Änderungen gehen verloren, wenn du das Projekt nicht exportierst.

ABBRECHEN

BESTÄTIGEN

1.1.3 Hilfe

- Dokumentation
- Datenschutzrichtlinie
- Impressum

1.2 Blaue Leiste (02)



1.2.1 Programm starten (1)

Der Code wird an den Controller übergeben und das Programm startet

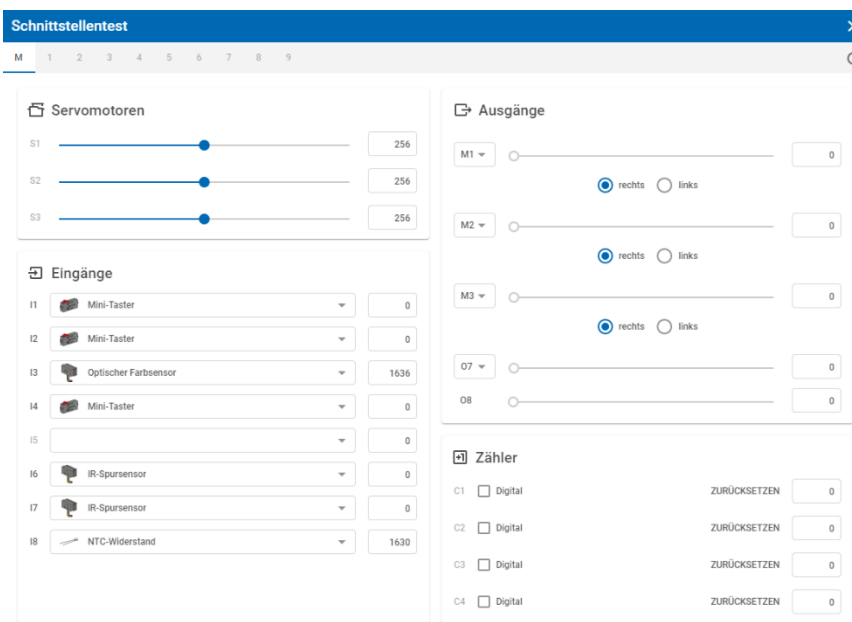
1.2.2 Programm beenden (2)

Das Programm auf dem Controller wird beendet

1.2.3 Debugging (3)

Das Programm wird auf Fehler überprüft und diese werden im Konsolenfenster (09) angezeigt

1.2.4 Schnittstellentest (4)



Alle angeschlossenen Aktoren und Sensoren, können auf ihre Funktionsfähigkeit getestet werden.

1.2.5 Programm hochladen (5)

Das Programm (xxx.py) wird auf den TXT 4.0 geladen und kann dann ohne Entwicklungsumgebung gestartet werden.

1.2.6 Controller verbinden (6)

Controller verbinden

IP-Adresse des Controllers *
192.168.178.60

API-Schlüssel *
XWHSpZ

Verbindung kann über WLAN, USB oder Bluetooth hergestellt werden. Der Schlüssel kann in Controller ausgelesen werden.

Im Erfolgsfall wird das Symbol mit einem grünen Punkt angezeigt.



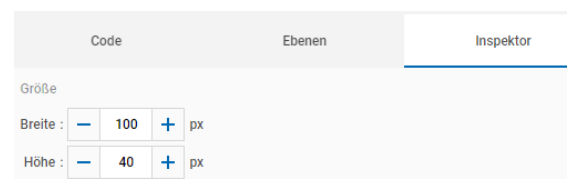
Weiterhin sind nun alle anderen Befehle aktiviert.

1.3 Rechte Seite (03)

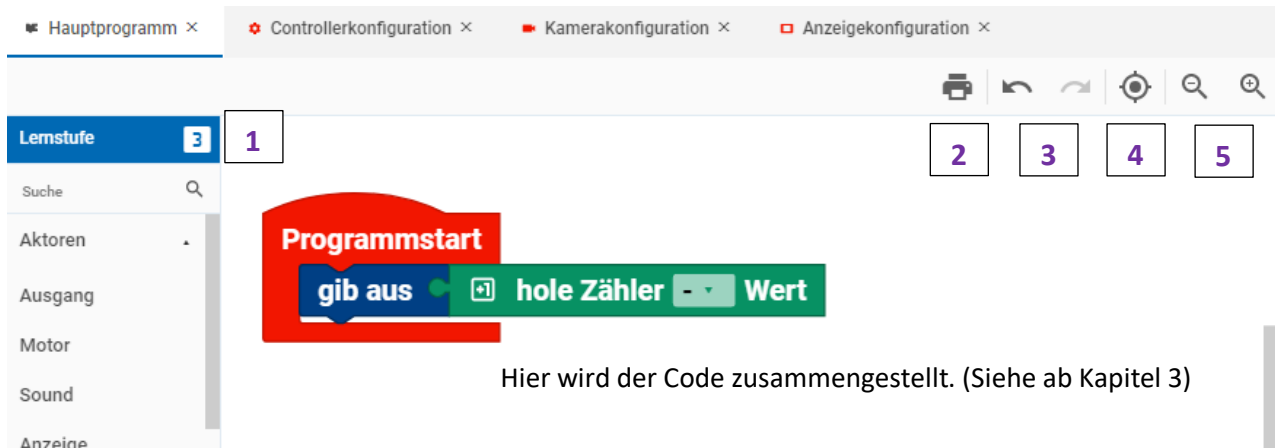
```
1 from lib.controller import *  
2  
3  
4 print(TXT_M_C1_motor_step_counter.get_count())  
5
```

Diese Anzeige ändert sich entsprechend der Auswahl des Fensters. Im Hauptprogramm und der Controllerkonfiguration wird der Python-Code angezeigt. In der Kamera- und Anzeigekonfiguration, ist der dritte

Punkt (Inspektor) wichtig. Hier werden die Eigenschaften des ausgewählten Elementes, wie z.B. des OK-Buttons eingestellt. Ligen bei der Kamerakonfiguration mehrere Sensorflächen übereinander, können unter dem Punkt Ebenen, die einzelnen Flächen ein- bzw. ausgeschaltet werden.

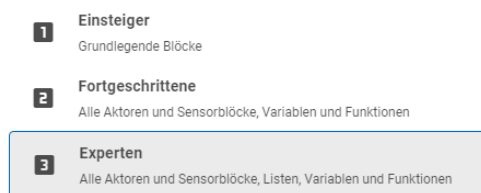


1.4 Hauptprogramm (04)



1.4.1 Lernstufen (1)

Lernstufe wählen



Es können unterschiedliche Programmier-Lernstufen eingestellt werden. Je nach benötigtem Schwierigkeitsgrad kann zwischen 3 Lernstufen ausgewählt werden.

1.4.2 Drucken (2)

Hierbei wird nur der Quellcode (das Programm) ausgedruckt.

1.4.3 Schritt zurück/vor (3)

Der jeweils letzte Schritt kann zurückgenommen oder wieder benutzt werden.

1.4.4 Zentrieren (4)

Der Code wird in die Mitte des Bildschirms gerückt.

1.4.5 Verkleinern/ Vergrößern (5)

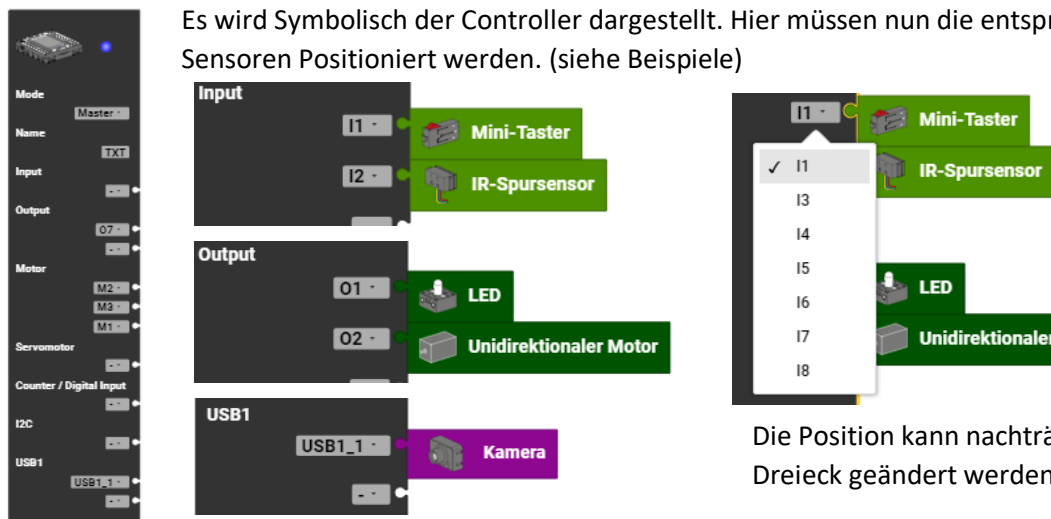
Der gesamte Quellcode wird verkleinert oder vergrößert dargestellt.

Tip: Einzelne Programmgruppen können auch zusammengeklappt werden, um die Übersichtlichkeit zu verbessern. Dazu rechte Maustaste auf Objektgruppe und die Option „Baustein zusammenfallen“ wählen.



1.5 Controllerkonfiguration (05)

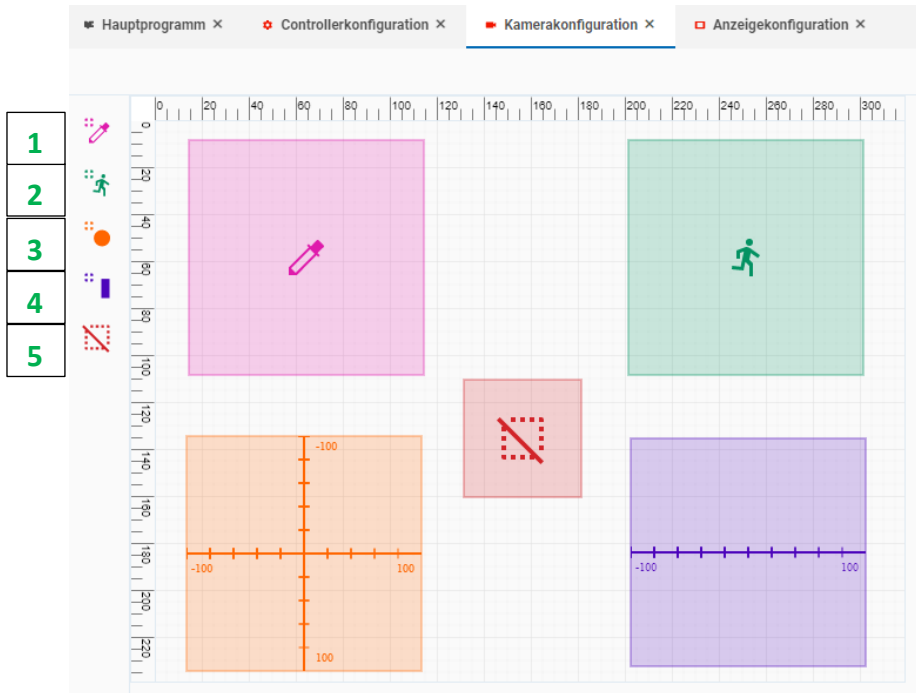
Es wird Symbolisch der Controller dargestellt. Hier müssen nun die entsprechenden Aktoren und Sensoren Positioniert werden. (siehe Beispiele)



Die Position kann nachträglich über das kleine Dreieck geändert werden.

1.6 Kamerakonfiguration (06)

Anwendung unter 4.4



Diese

Größe

Breite : px

Höhe : px

Position

x : px

y : px

Elemente sind im Inspektor, für alle Elemente gleich. Jedes Element hat dazu noch individuelle Einstellmöglichkeiten.

1.6.1 Farberkennung (1)

FARBERKENNUNG							
Name	Rot	Grün	Blau	Hex	Hue	Saturation	Wert
color_detector	118	115	107	76736b	43	9	46

Zur Kontrolle der Werte, gibt es eine Testumgebung unter der Konfigurationseinstellung im Punkt „camera“.

Farbe

Kontrast :

Der Inspektor wird um den Punkt Kontrast erweitert.

Der Abruf im Hauptprogramm erfolgt unter dem Punkt:

USB

1.6.2 Bewegungserkennung (2)

BEWEGUNGSERKENNUNG	
Name	Bewegung
motion_detector	false

Wird im aufgezogenen grünen Fenster eine Bewegung erkannt, ändert sich die Eigenschaft von motion_detector auf True.

Der Inspektor wird um den Punkt Toleranz erweitert:

Bewegung

Toleranz :

1.6.3 Ballerkennung (3)

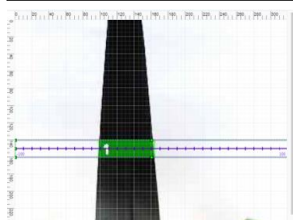
BALLERKENNUNG			
Name	X	Y	Durchmesser
ball_detector			

Der Inspektor wird um folgende Punkte erweitert:

In der Testumgebung wird die Position und der Durchmesser angezeigt.

1.6.4 Linienerkennung (4)

LINIENERKENNUNG						
Name	Linie	Position	Breite	Rot	Grün	Blau
line_detector	1	-30	10	0	0	0



Hier wird die Abweichung von der Mitte des Erkennungsfensters erkannt.

Anzahl der Linien

Anzahl :

Linienbreite

Minimum :

Maximum :

Bereich

Start :

Ende :

Durchmesser

Minimum :

Maximum :

Bereich x-Achse

Start :

Ende :

Farbe

Farbe :

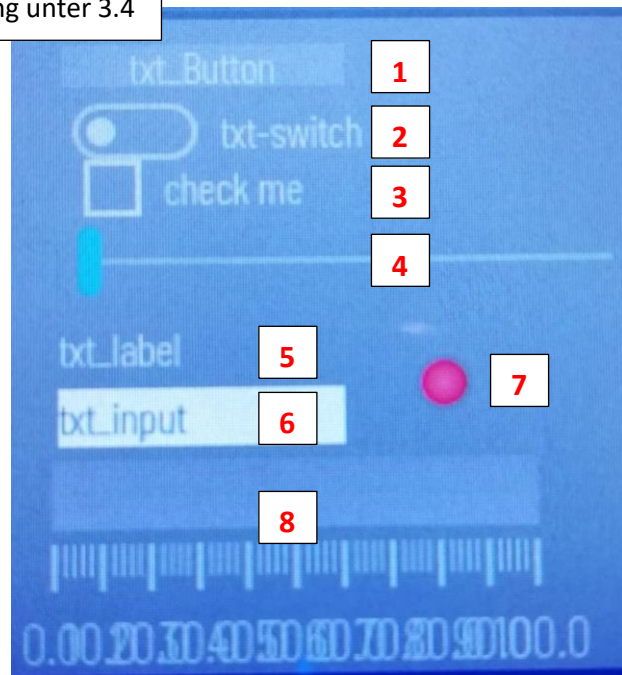
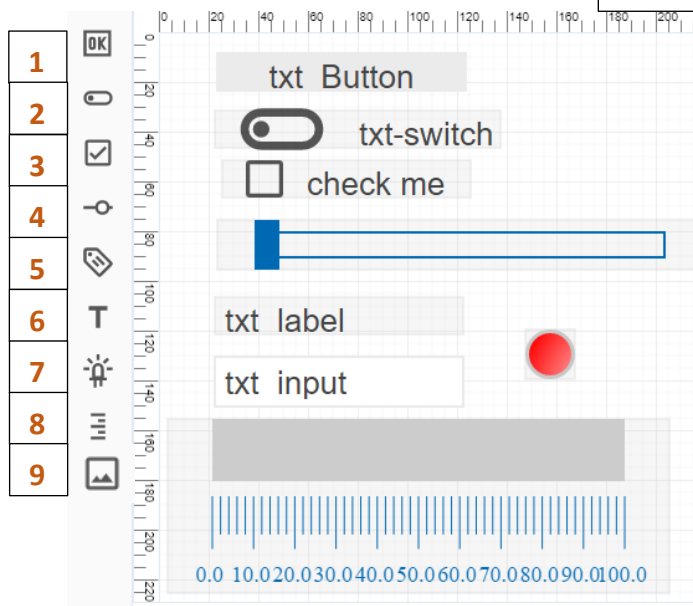
Hue-Toleranz : Grad

1.6.5 Aussparung (5)

Dieses Element dient dazu, Bereiche aus dem Erkennungsbereich auszublenden.

1.7 Anzeigekonfiguration (07)

Anwendung unter 3.4



1.7.1 Inspektor

Hier können die Eigenschaften der Touchscreen Elemente verändert werden.

Gemeinsam haben alle die Möglichkeit der Veränderung von Größe, Position, Name und Text. Daneben gibt es noch individuelle Einstellmöglichkeiten.

1.7.2 Button (1)

Mit Text belegbarer Knopf, dessen Status ausgewertet werden kann. Änderung der Textgröße siehe Punkt 5.4.9

1.7.3 Schiebeschalter (2)

Schiebeschalter, dessen Status ausgewertet werden kann.

1.7.4 Kontrollfeld (3)

Checkbutton zur Anzeige, aber auch zur Steuerung von Aktivitäten.

1.7.5 Schieberegler (4)

Für z.B. Servo, Motoren oder Lampenhelligkeit. Regelbereich kann im Inspektor verändert werden.

1.7.6 Label (5)

Einfache Textausgabe. Änderung der Textgröße siehe Punkt 5.4.9

1.7.7 Input (6)

Feld zur Eingabe von Text, der im Programm ausgewertet werden kann.

1.7.8 Status Indikator (7)

Anzeige kann gesetzt bzw. ausgelesen werden

1.7.9 Messinstrument (Gauge) (8)

Zur Anzeige analoger Werte. Der Messbereich kann im Inspektor verändert werden.

1.7.10 Bilddarstellung (9)

Es besteht die Möglichkeit, Bilder vom Rechner auf der Oberfläche darzustellen.

Größe

Breite : px

Höhe : px

Position

x : px

y : px

Identität

Name :

TXTButton

Text :

Gauge

Anzahl Unter-Tickmarks :

Tickmark-Schrittweite :

Ausrichtung: ☒ horizontal ☐ vertical

Von :

Wert :

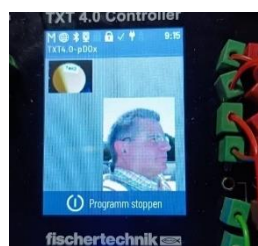
Bis :

Identität

Name :

TXImage

Bild :



Button

TXUIButton

Text :

Aktiviert : ☒

Schiebeschalter

TXTSwitch

Text :

Aktiviert : ☒

Angeklickt : ☐

Schieberegler

TXTSlider

Aktiviert : ☒

Ausrichtung: ☒ horizontal
☐ vertical

Von :

Wert :

Bis :

Status Indikator

StatusIndicator

Aktiv : ☒

Farbe :

1.8 Bedienfeld

Das nachträglich ins Programm implementierte Bedienfeld wird unter Punkt 6.1 ausführlich beschrieben.

1.9 Konsole (09)

Im Gegensatz zu „Anzeige“, wo Werte auf dem Controller-Display erscheinen, werden hier Informationen auf dem Rechner im Fenster Konsole ausgegeben. Das können zum einen die Info über Start und Ende eines Programmes sein. Aber auch Werte zu Kontrolle oder Fehlermeldungen werden hier angezeigt.



1.10 Haltepunkte (10)

Noch nicht Dokumentiert.

1.11 Ausdrucke (11)

Noch nicht Dokumentiert.

1.12 Aufrufstapel (12)

Noch nicht Dokumentiert.

2 Informationen zum Umgang mit Blöcken

2.1 Arten der Blöcke

Zu einem Objekt (hier eine LED) gibt es verschiedene Blöcke zum Umgang mit dem Wert oder Ereignis. Erkennbar ist dies an der Art der Darstellung auf der linken Seite.



Startblock

Block zum Setzen von Werten

Block zum Lesen von Werten

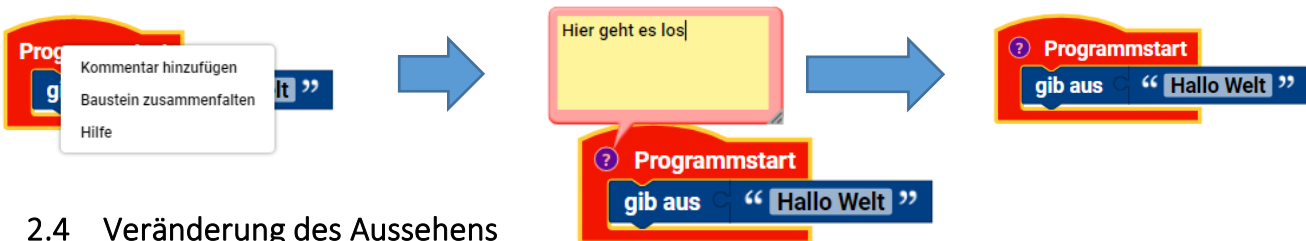
Block zum Vergleichen von Werten

2.2 Block vervielfältigen

Mit Klick der rechten Maustaste auf den Block eröffnet sich die Möglichkeit diesen Baustein zu kopieren oder zu duplizieren. Duplizieren geht schneller, da der neue Block sofort in der Nähe angelegt wird. Kopieren eignet sich besser, wenn der neue Block weit entfernt „Eingefügt“ wird.

2.3 Kommentar hinzufügen

Damit später ein Programm nachvollziehbar ist, gibt es die Möglichkeit, den Blöcken Kommentare zuzufügen. Mit Klick der rechten Maustaste auf den Block kann der Punkt „Kommentare hinzufügen“ benutzt werden. Durch einen Klick auf das Fragezeichen wird die Kommentarblase geöffnet bzw. geschlossen.



2.4 Veränderung des Aussehens

Es gibt einige Blöcke, bei denen man die Darstellung verändern kann. Mit Klick der rechten Maustaste auf den Block kann der Punkt „externe/interne Eingänge“ benutzt werden.



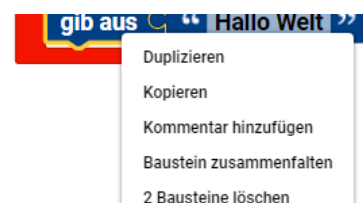
2.5 Verkleinerte Darstellung

Um die Übersicht der Programme zu verbessern, gibt es die Möglichkeit, die Blöcke verkleinert darzustellen. Mit Klick der rechten Maustaste auf dem Block kann der Punkt „Baustein zusammenfallen“ benutzt werden.



2.6 Baustein löschen

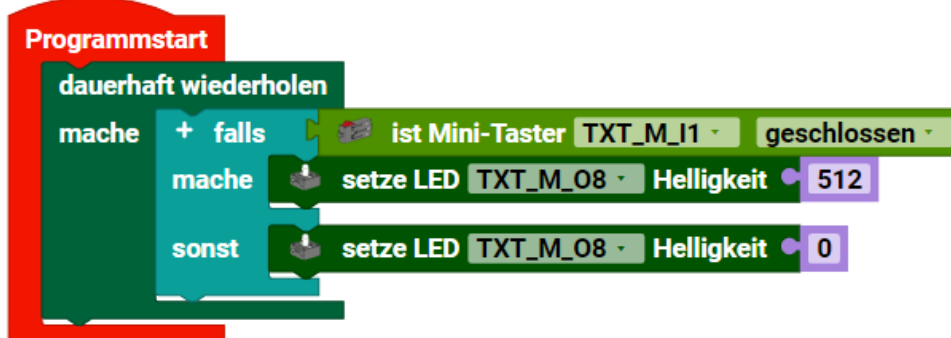
Das Löschen der Blöcke erfolgt ebenfalls im Kontextmenü des Blockes.



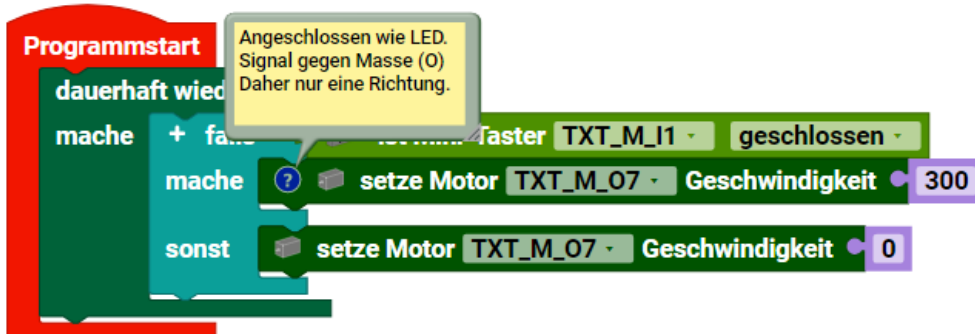
3 Aktoren

3.1 Ausgang

3.1.1 LED



3.1.2 Motor



3.1.3 Magnetventil



Anschluss wie unidirektionale Motoren.

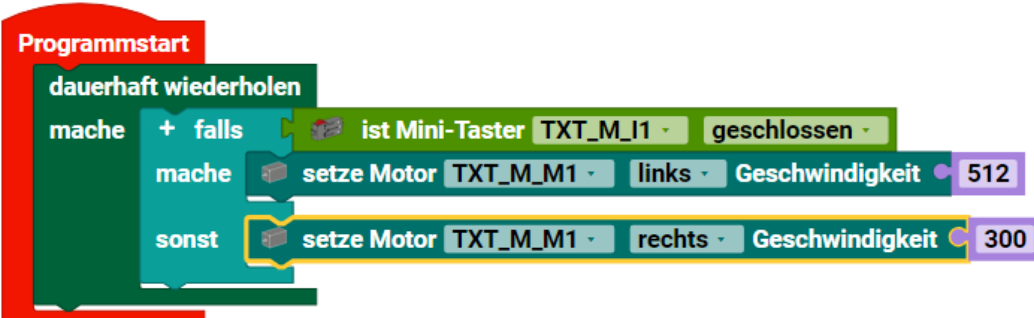
3.1.4 Kompressor



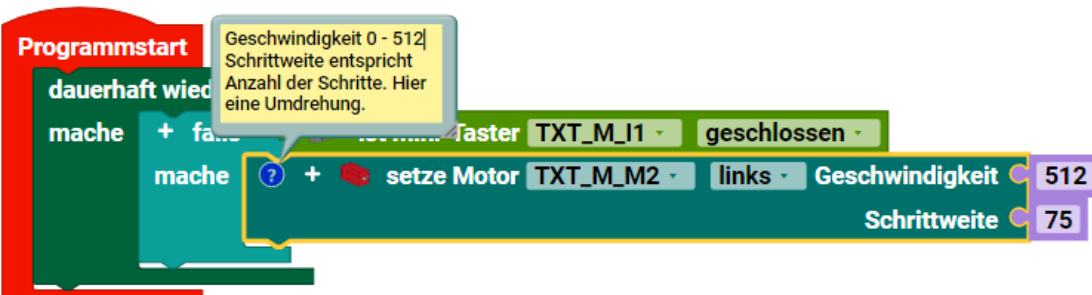
Anschluss wie unidirektionale Motoren.

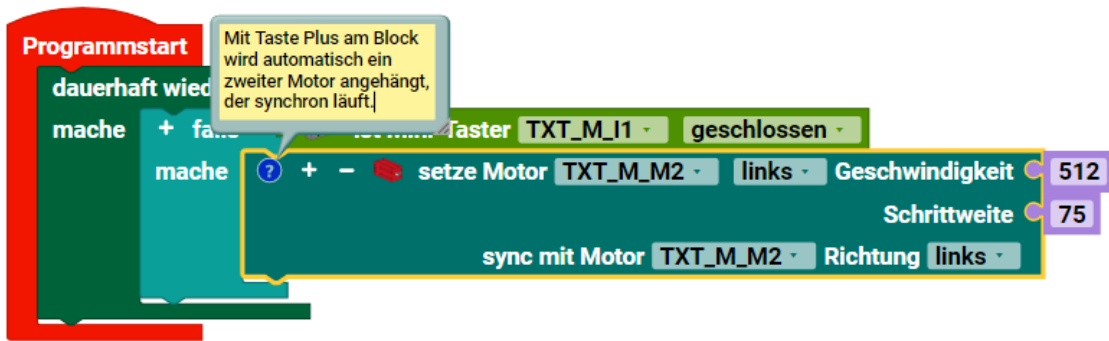
3.2 Motor

3.2.1 Motor (Bidirektional)

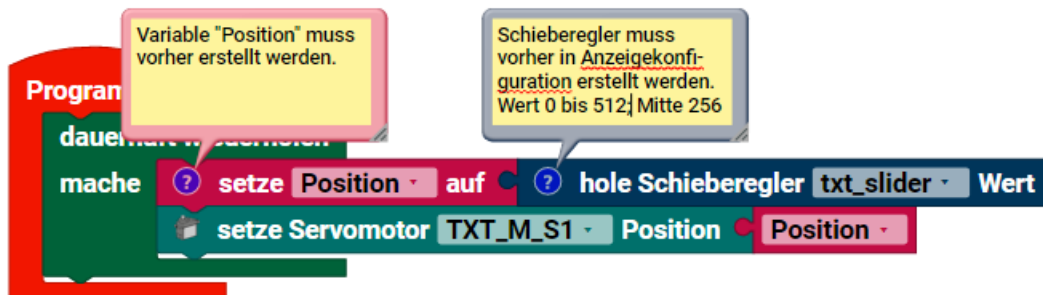


3.2.2 Encoder Motor

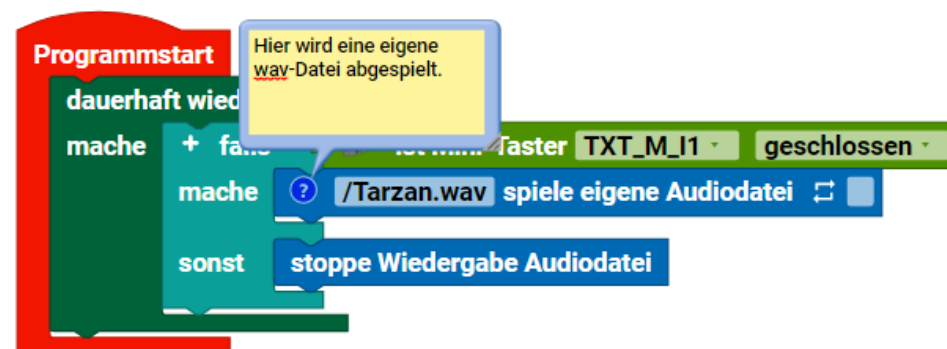
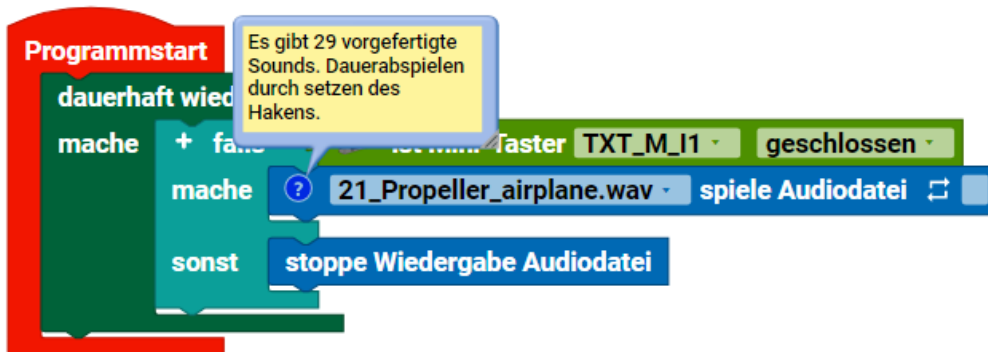




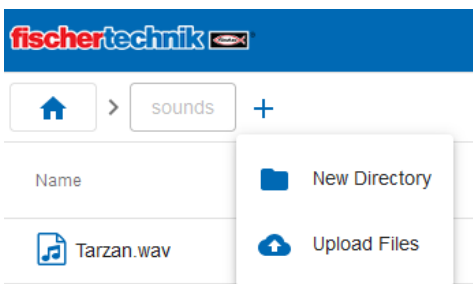
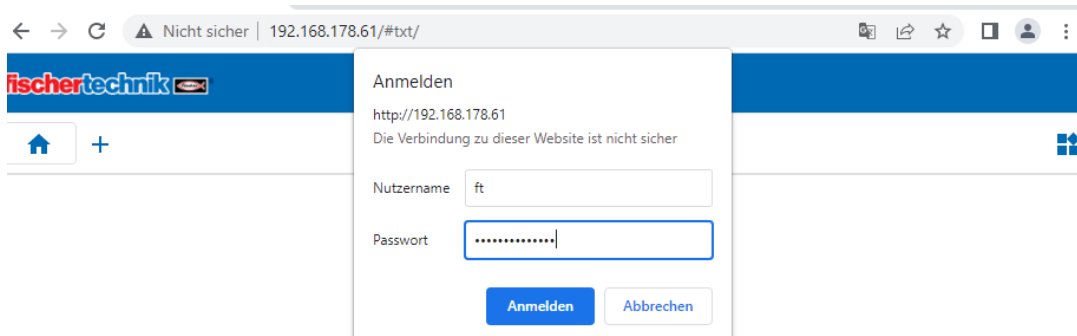
3.2.3 Servomotor



3.3 Sound



Für eigene Wav-Dateien muss man sich mit dem Controller verbinden. Aufruf der IP-Adresse des Controllers.
Nutzername: ft und Passwort: fischertechnik eingeben.



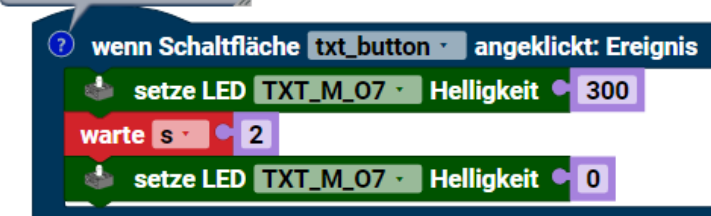
- Den Ordner Sounds bzw. Geräusche öffnen.
- Plustaste drücken.
 - Upload Files.
 - Add Files und die entsprechende Datei auswählen.
- Upload-Knopf drücken.
- Finish-Knopf drücken
- Abruf siehe Beispiel: /Tarzan.wav

3.4 Anzeige

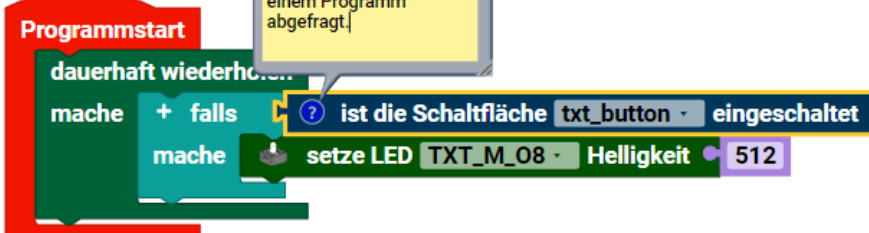
Für das jeweilige Element muss vorher das Objekt im Anzeige Konfigurator erstellt werden. Im Inspektor werden dann die Eigenschaften festgelegt.

3.4.1 Button (txt_button)

Wird Schaltfläche auf dem Display gedrückt, Leuchtet die LED für 2 Sekunden.

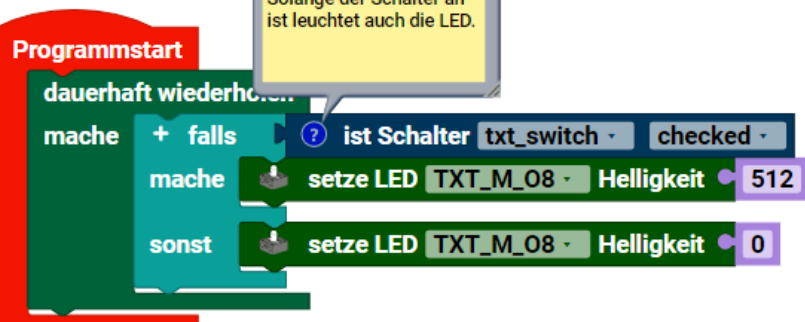


Hier wird der Button in einem Programm abgefragt.



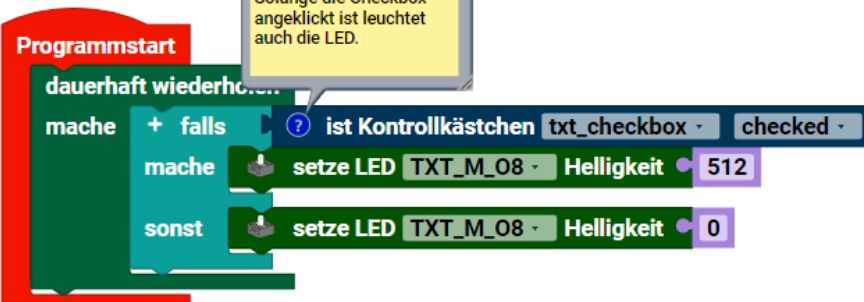
3.4.2 Schalter (txt_switch)

Solange der Schalter an ist leuchtet auch die LED.

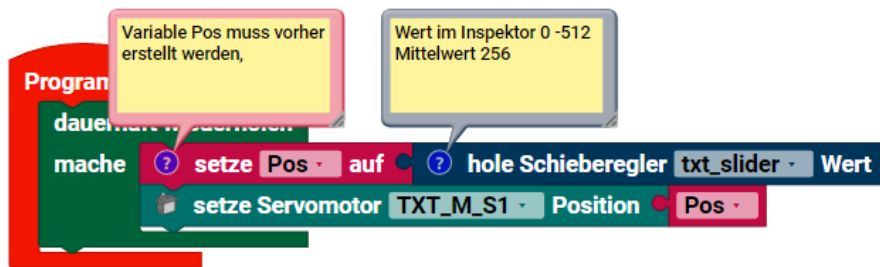


3.4.3 Kontrollfeld (txt_checkbox)

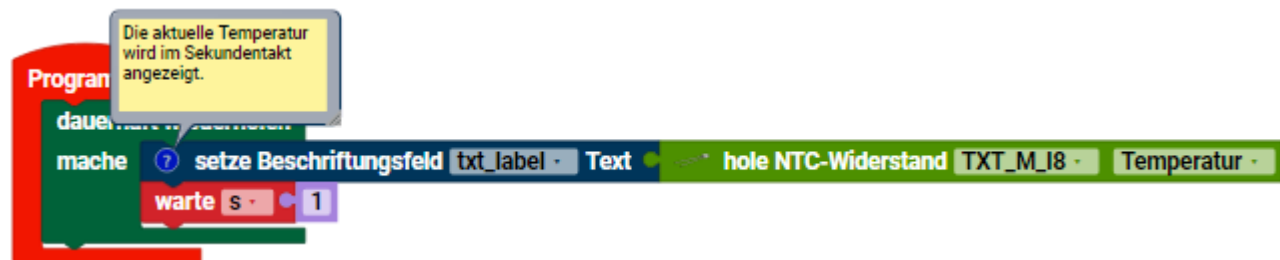
Solange die Checkbox angeklickt ist leuchtet auch die LED.



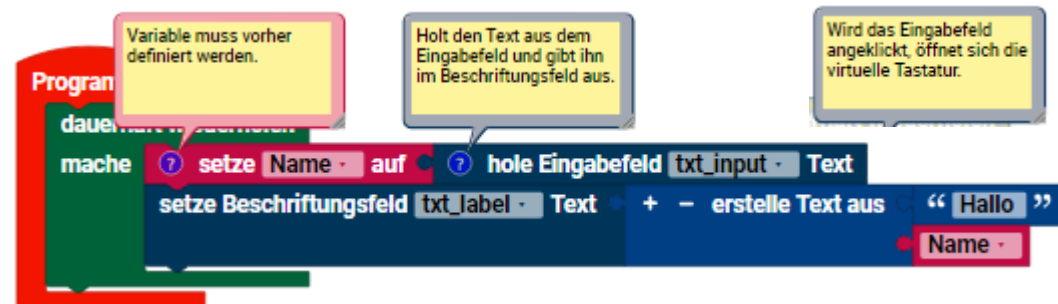
3.4.4 Schieberegler (txt_slider)



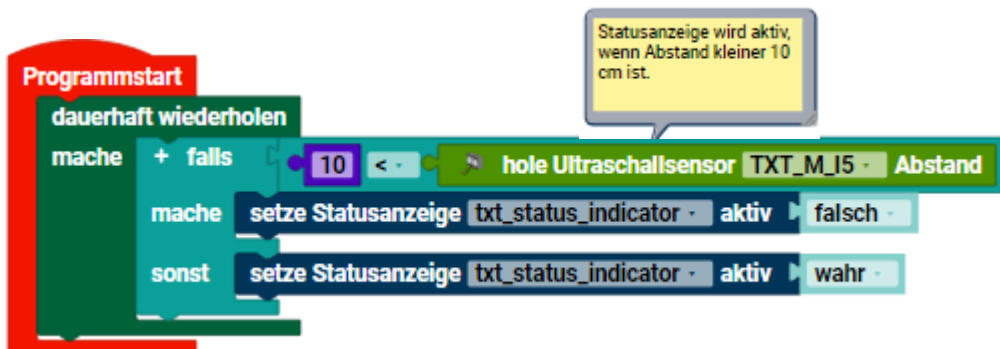
3.4.5 Beschriftungsfeld (txt_label)



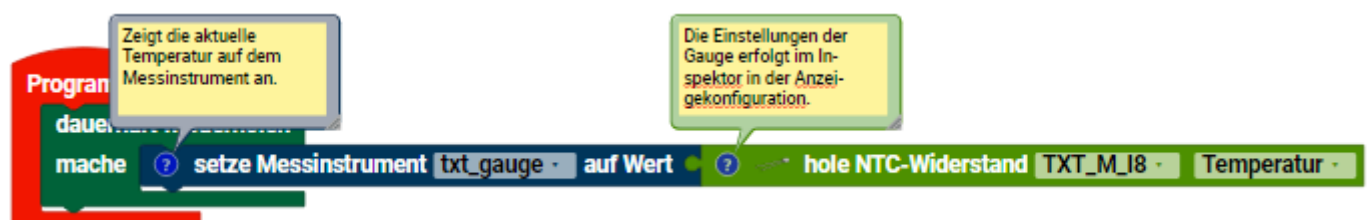
3.4.6 Eingabefeld (txt_input)



3.4.7 Statusanzeige (txt_status_indicator)



3.4.8 Messinstrument (txt_gauge)



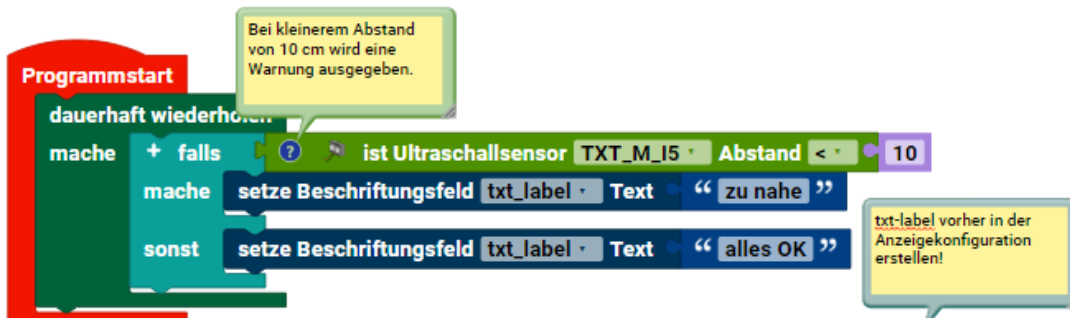
4 Sensoren

4.1 Eingang

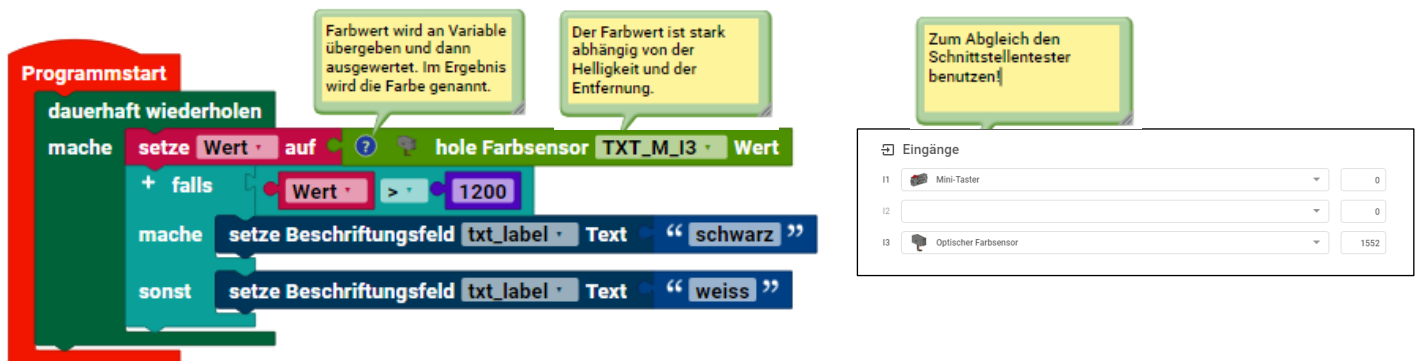
4.1.1 Taster



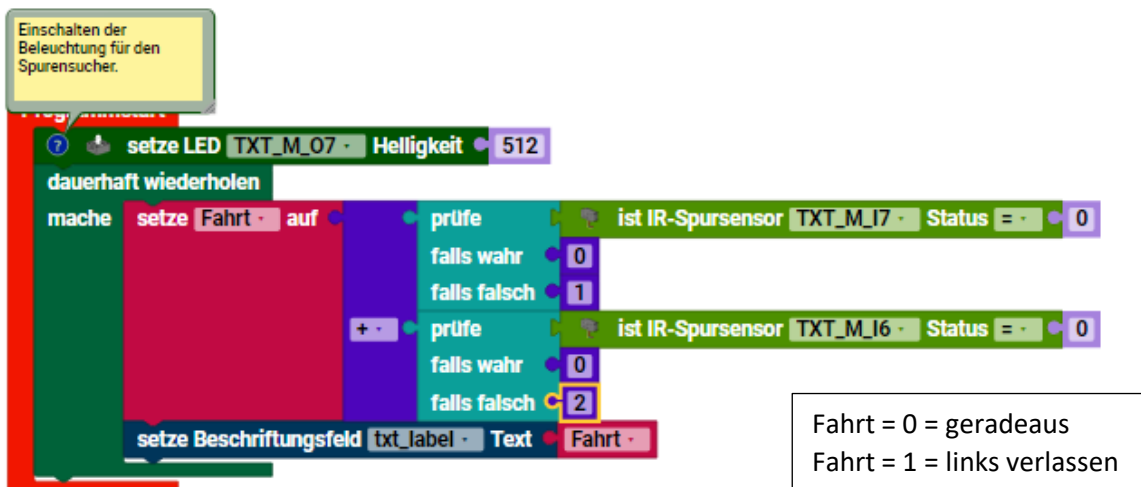
4.1.2 Ultraschallsensor



4.1.3 Farbsensor



4.1.4 Spurensucher

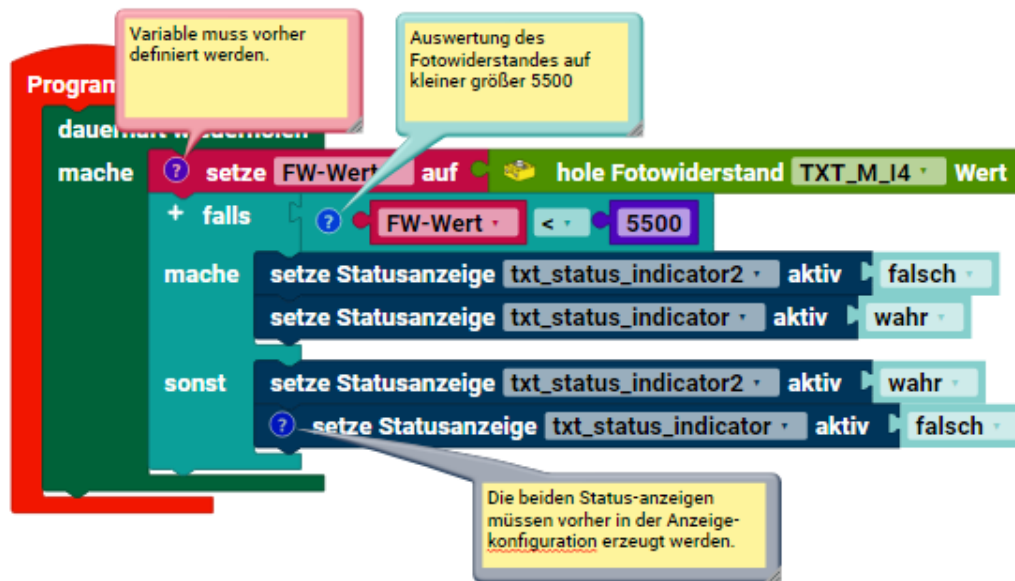


Fahrt = 0 = geradeaus
Fahrt = 1 = links verlassen
Fahrt = 2 = rechts verlassen
Fahrt = 3 = Spur verloren

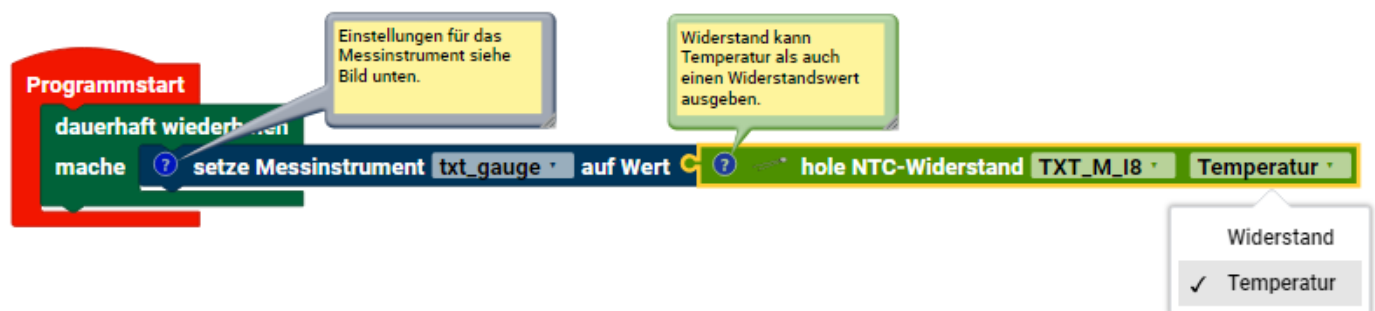
4.1.5 Fototransistor



4.1.6 Fotowiderstand



4.1.7 NTC Widerstand



Gauge

Anzahl Unter-Tickmarks:

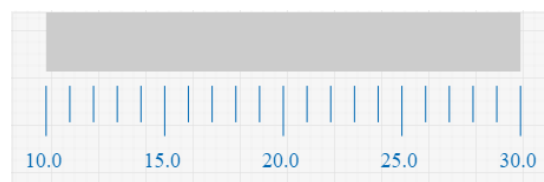
Tickmark-Schrittweite:

Ausrichtung: ☒ horizontal ☐ vertical

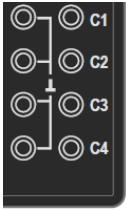
Von:

Wert:

Bis:



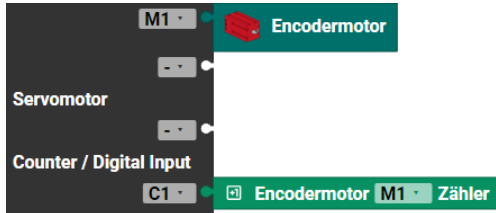
4.2 Zähler



Impulse können auf zwei verschiedene Arten gezählt werden:

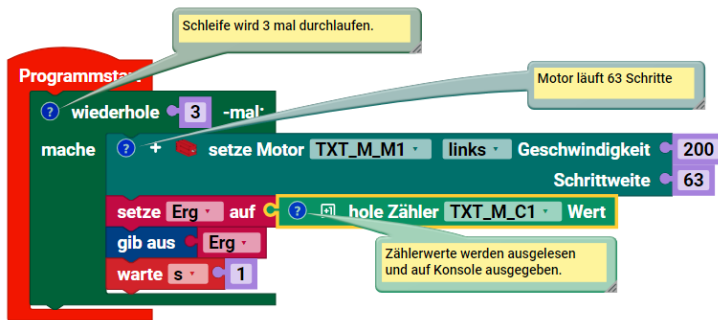
- Die Auswertung erfolgt über einen normalen I-Eingang
- Bei schnellen Impulsen würde ein C-Eingang (C1 – C4) besser funktionieren.
- Hauptsächlich gedacht sind die C-Eingänge für Encodermotoren, die über 3 zusätzliche Anschlüsse verfügen (Plus, Minus, Zählausgang)

Bei den aktuellen Motoren beträgt eine Umdrehung 63,3 Impulse.(Alt-63,9)



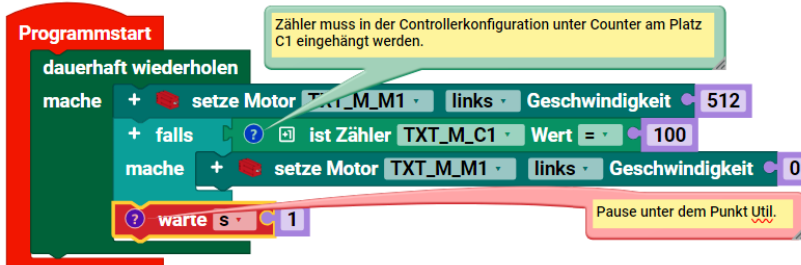
Um die Zählimpulse eines Motors auszuwerten, muss in der Controllerkonfiguration der Motor und der entsprechende Zähleingang am Controller angemeldet sein.

4.2.1 Hole Zählerwert



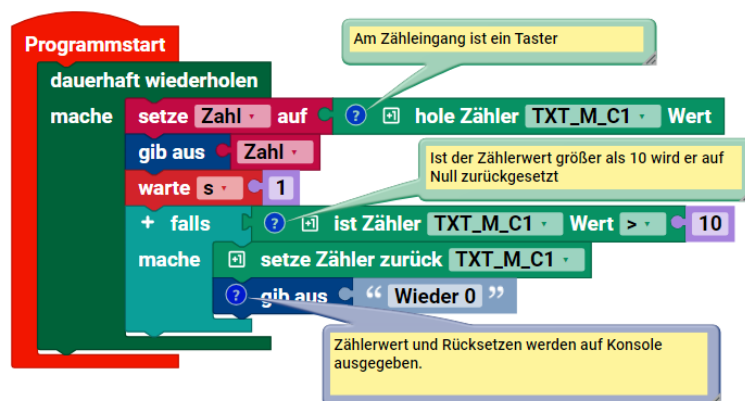
Konsole
Program startet...
63
63
63

4.2.2 Ist Zähler ... Wert



Ist die Zahl größer 100 wird der Motor Angehalten. Nach einer Pause startet der Motor erneut.







4.2.3 Setze Zähler zurück



Konsole
Program startet...
0
0
4
6
8
10
Wieder 0
18
0
5
6
10
Wieder 0
16
0
0

4.3 I²C

Hier können die 3 Sensoren von Fischertechnik angeschlossen werden. Das sind zurzeit:

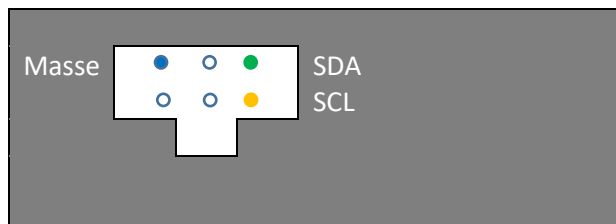
2 mal 10 Polig für TXT	1 mal 6 polig für TXT 4.0	Beschreibung
 158402	 201257	Kombisensor mit Gyroskop Beschleunigungssensor Kompasssensor
 167358	 182974	Umweltsensor mit Temperatur Luftfeuchtigkeit Luftdruck Luftqualität
 186705	 183267	Gestensensor mit Dämmerungsschalter Farberkennung Licht und Nähe Gestensteuerung

Es gibt die Sensoren teilweise für den alten TXT bzw. für den neuen TXT 4.0 Sie sind vom Anschluss nicht Kompatibel. Dafür bietet FT einen Adapter für über 20 Euro an.

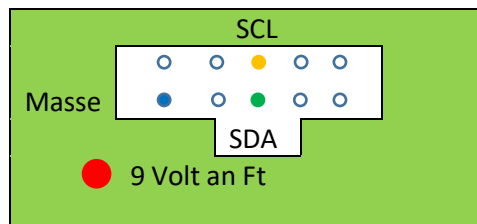


Alternativ braucht man nur ein Kabel im Eigenbau mit folgender Belegung:

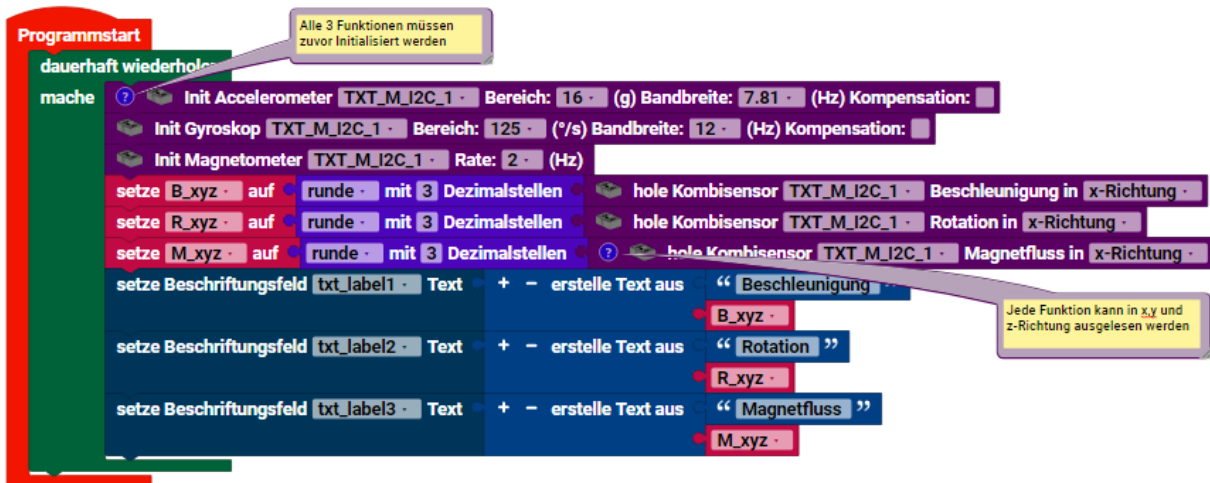
Ft TXT 4.0 Ansicht auf Anschlussbuchse



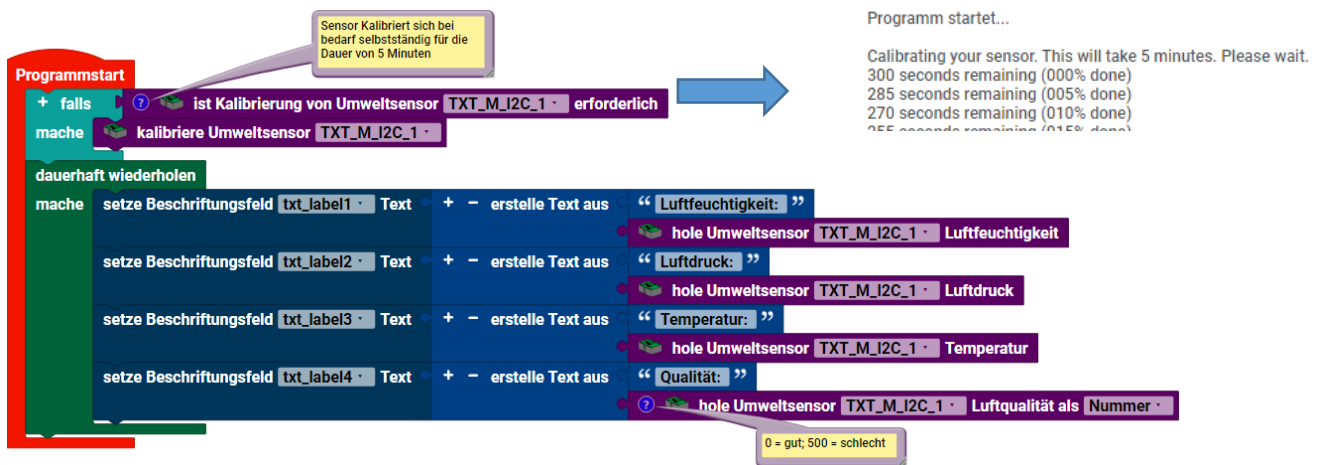
Ft Umweltsensor Ansicht auf Anschlussbuchse



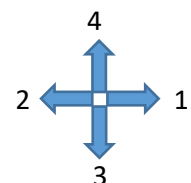
4.3.1 Kombisensor



4.3.2 Umweltsensor



4.3.3 Gestensensor



4.4 USB (Kamera)

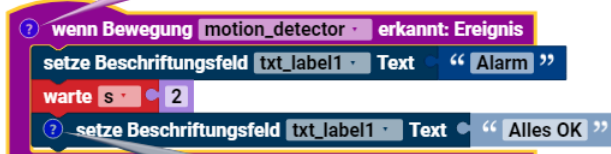
4.4.1 Bewegungserkennung

Dieser Block muss immer im Programm enthalten sein, auch wenn die Abfrage über ein anderes Startfeld abgearbeitet wird.



Auch die Dauerschleife ist zwingend notwendig.

Das Bewegungsfeld muss in der Kamerakonfiguration definiert sein



Feld muss im Anzeigeconfigurator definiert sein.



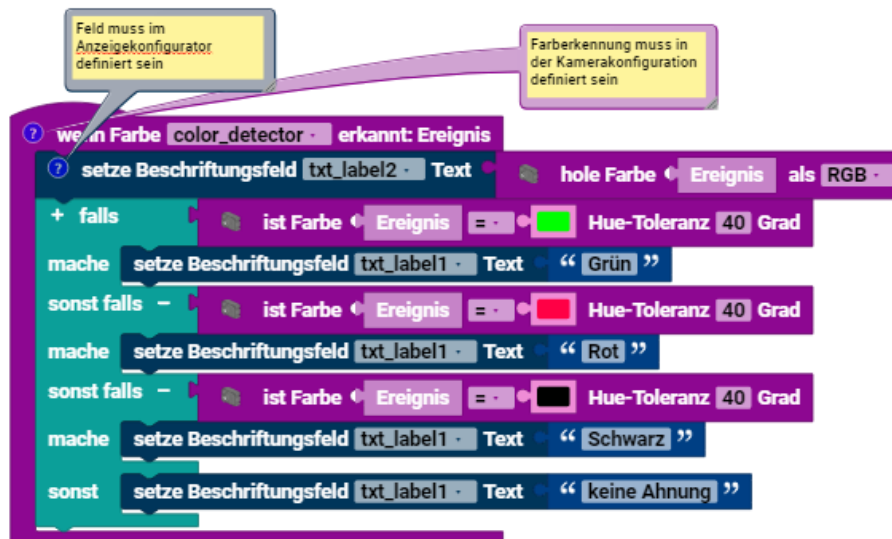
4.4.2 Farberkennung



Die Werte können vorher in der Kamerakonfiguration unter Camera kontrolliert werden

FARBERKENNUNG							
Name	Rot	Grün	Blau	Hex	Hue	Saturation	Wert
color_detector	164	128	131	a48083	355	21	64

Hue – Farbtön
Saturation - Farbsättigung



Farberkennung muss in der Kamerakonfiguration definiert sein

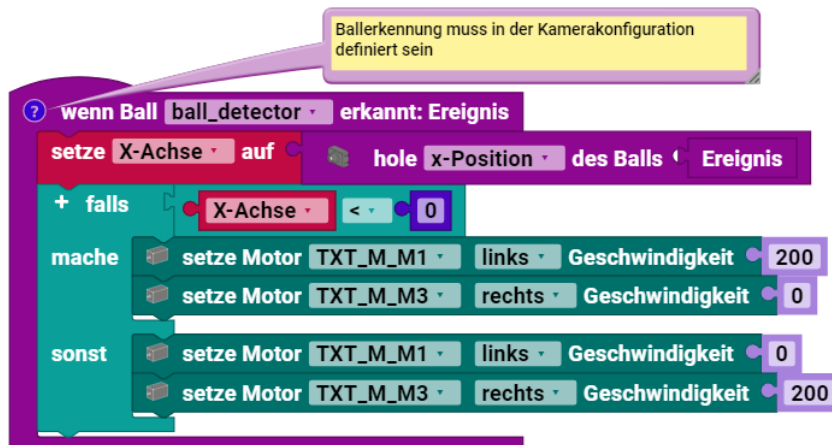
Feld muss im Anzeigeconfigurator definiert sein

4.4.3 Ballerkennung



Die Werte können vorher in der Kamerakonfiguration unter Camera kontrolliert werden

BALLERKENNUNG				
Name	X	Y	Durchmesser	
ball_detector	-16	25	50	



Ballerkennung muss in der Kamerakonfiguration definiert sein

4.4.4 Linienerkennung

Variablen pos und breite müssen vorher angelegt werden.

Das Linienfeld muss in der Kamerakonfiguration definiert sein

Beide Werte werden in den Beschriftungsfeldern ausgegeben



LINIENERKENNUNG						
Name	Linie	Position	Breite	Rot	Grün	Blau
line_detector	1	-21	44	0	0	0

Die Werte können vorher in der Kamerakonfiguration unter Camera kontrolliert werden



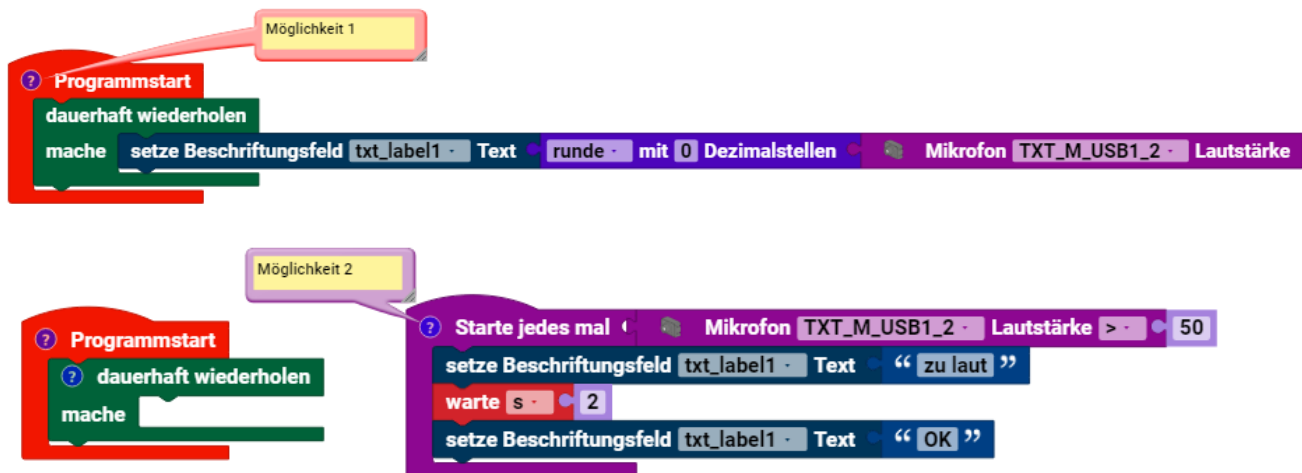
Feld muss im Anzeigeconfigurator

4.4.5 Bilddarstellung auf Controller



* Base64 ist eine Möglichkeit, Binärdaten in einer ASCII-Zeichenfolge darzustellen. Das bedeutet, dass ein Bild im Browser nicht über eine Binärdatei wie eine JPG- oder GIF-Bild eingebunden wird, sondern der Browser aus einer Zeichenfolge das Bild direkt berechnet.

4.4.6 Mikrophon



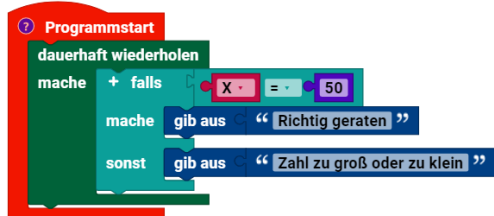
5 Verarbeitung

5.1 Logik

5.1.1 Boolesche Logik

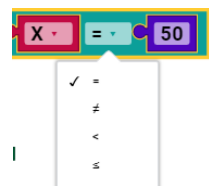
Einfaches mathematisches System, das zwei Werte hat, um Bedingungen und Schleifen zu kontrollieren. (wahr,falsch)

5.1.2 Wenn Funktion



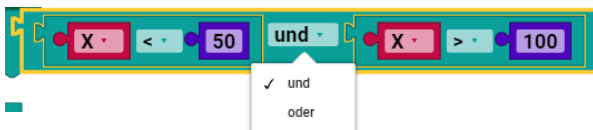
Es ist eine der häufigsten verwendeten Funktionen. Sie ermöglicht den logischen Vergleich zwischen einem aktuellen Wert und einem erwarteten Wert.

5.1.3 Vergleichsoperatoren



Jedem der 6 möglichen Vergleichsoperatoren werden 2 Eingänge übergeben und der Vergleich gibt wahr oder falsch zurück.

5.1.4 Logische Operatoren

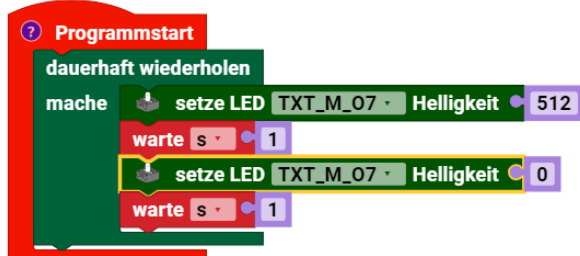


Der **und**-Block gibt dann und nur dann **wahr** zurück, wenn seine beiden Eingangswerte wahr sind.

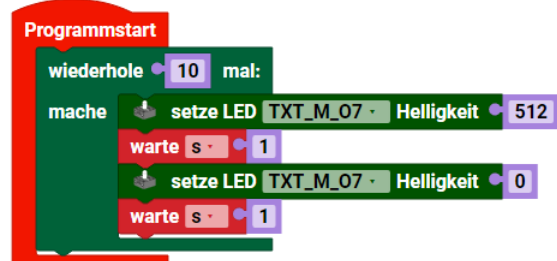
Der **oder**-Block gibt **wahr** zurück, wenn mindestens einer seiner beiden Eingangswerte wahr ist.

5.2 Schleifen

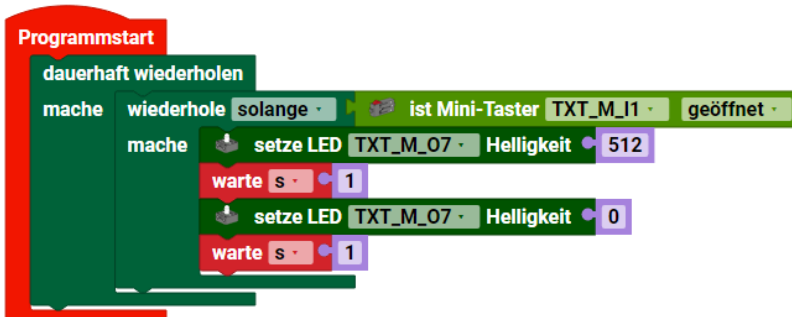
5.2.1 Dauerhafte Wiederholung (Dauerschleife)



5.2.2 Wiederhole (Zählschleife)

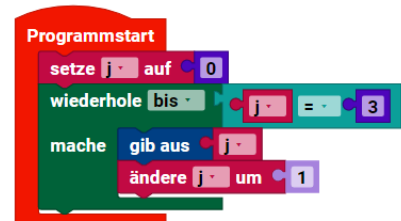


5.2.3 Wiederhole solange (Kopfschleife)

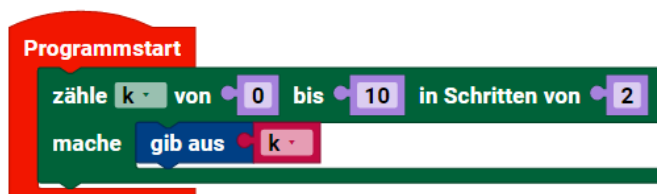


5.2.4 Wiederhole bis (Fußschleife)

Schleife wird mindestens 1 mal durchlaufen



5.2.5 Zählschleife von bis (For-Schleife)

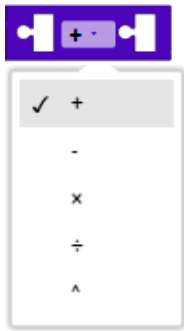


Die Variable „k“ wird wie folgt ausgegeben:

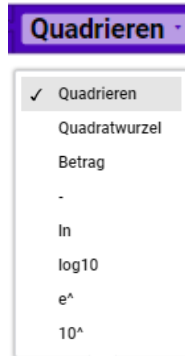
0, 2, 4, 6, 8, 10

5.3 Mathe

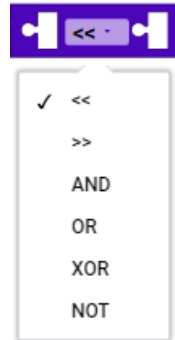
5.3.1 Einfache Rechnungen



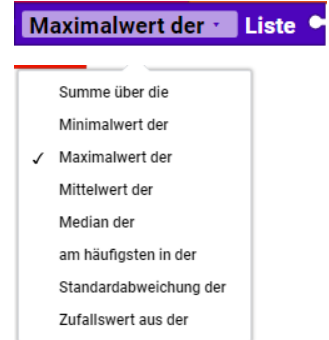
5.3.2 Spezielle Rechnungen



5.3.3 Operatoren *1



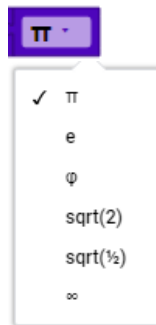
5.3.4 Auswertung von Listen *2



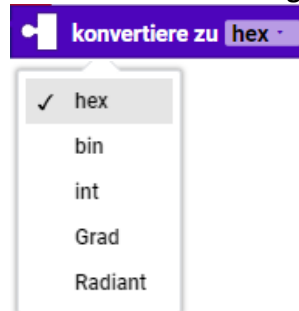
5.3.5 Funktionen



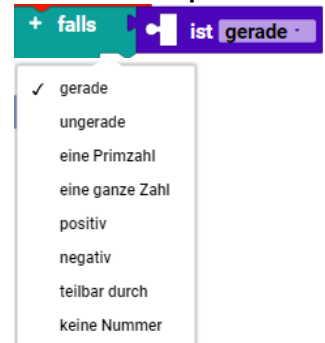
5.3.6 Konstanten



5.3.7 Konvertierung



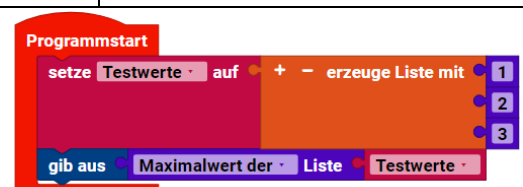
5.3.8 Zahl prüfen



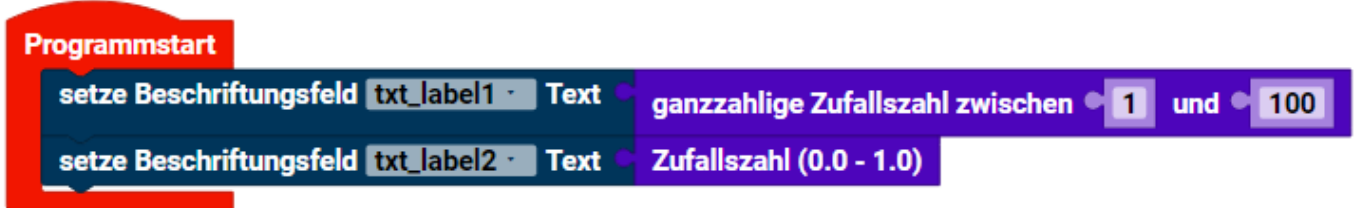
*1 Die „Bitweisen Operatoren“ sind für Ganzzahlen, die in das Binärformat umgewandelt werden und dann auf Bit Ebene je nach Operator „behandelt“ werden.

Bitzahl nach links schieben	Bitzahl nach rechts schieben	Bitzahlen mit UND vergleichen									
Die Zahl 2 (0010) wird zweimal nach links verschoben 0010 = 2 (Ausgangswert) 0100 = 4 (erste Verschiebung) 1000 = 8 (zweite Verschiebung)	Die Zahl 9 (1001) wird zweimal nach rechts verschoben 1001 = 9 (Ausgangswert) 0100 = 4 (erste Verschiebung) 0010 = 2 (zweite Verschiebung)	Potenz	2 ²	2 ¹	2 ⁰						
		Wert	4	2	1						
		5	1	0	1						
		3	0	1	1						
		1	0	0	1						
Bitzahlen mit ODER vergleichen	Bitzahlen mit XOR vergleichen	Bitzahlen mit NICHT vergleichen									
Potenz	2 ²	2 ¹	2 ⁰	Potenz	2 ²	2 ¹	2 ⁰	Potenz	2 ²	2 ¹	2 ⁰
Wert	4	2	1	Wert	4	2	1	Wert	4	2	1
5	1	0	1	5	1	0	1	3	0	1	1
3	0	1	1	3	0	1	1	-4	1	0	0
7	1	1	1	6	1	1	0				

*2 Voraussetzung für 5.3.4 arbeiten mit Listen



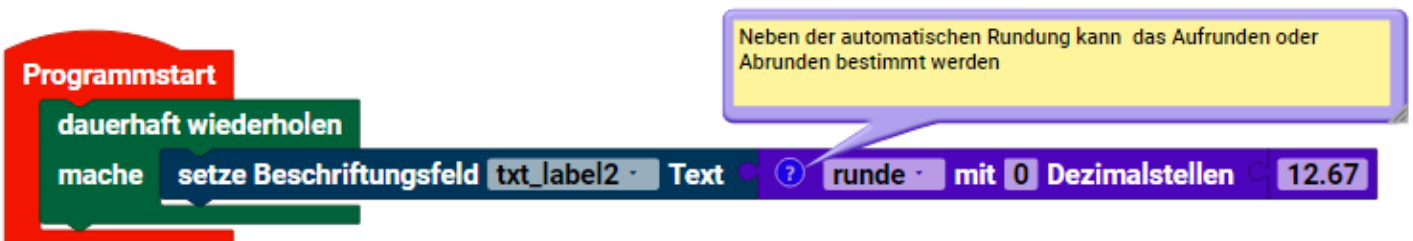
5.3.9 Zufallszahlen



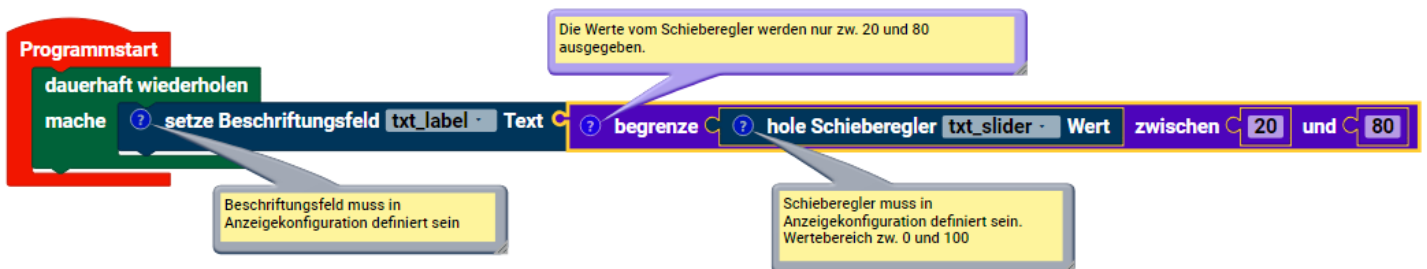
5.3.10 Rest einer Division



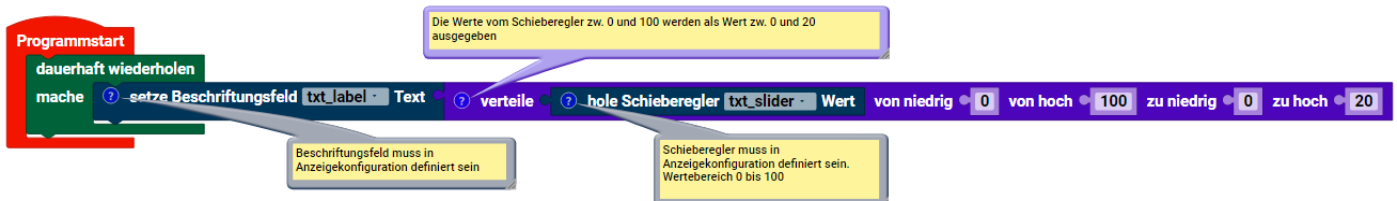
5.3.11 Zahl runden



5.3.12 Werte beschränken



5.3.13 Wertebereich wandeln



5.3.14 Atan2

Mit der Funktion wird der Arkustangens (auf Basis von XY-Koordinaten) der Pixel in einem Raster berechnet.



5.4 Text

Die Ausgabe von Texten erfolgt immer in der Konsole unter dem Quelltext. Sie können aber auch auf dem Touchfeld des Controllers angezeigt werden. Dazu muss ein entsprechendes Feld (Beschriftungsfeld, Eingabefeld) in der Anzeigekonfiguration definiert werden.

5.4.1 Textausgabe

Beispiel 1: Ein 'gib aus' Block mit dem Text 'Hallo Welt'. Die Konsole zeigt: Programm startet..., Hallo Welt, Programm beendet.

Beispiel 2: Ein 'gib aus' Block, der mit 'erstelle Text aus' verbunden ist. Dieser enthält die Teile 'Die ', '7' und ' Geislein '. Die Konsole zeigt: Programm startet..., Die 7 Geislein, Programm beendet.

Beispiel 3: Ein 'gib aus' Block, der mit 'setze Element auf' und 'zu Element Text anhängen' verbunden ist. Die Konsole zeigt: Programm startet..., Hello World alles OK., Programm beendet.

5.4.2 Textlänge ermitteln

Ein 'gib aus' Block mit dem Text 'Länge von' und dem Text 'Das ist ein kurzer Text.'. Die Konsole zeigt: Programm startet..., 24, Programm beendet.

5.4.3 Prüfen auf leeren Text

Ein 'falls' Block mit dem Text 'ist leer'. Wenn 'ja', wird 'gib aus' mit 'Kein Text vorhanden' ausgeführt. Wenn 'sonst', wird 'gib aus' mit 'Text vorhanden' ausgeführt. Die Konsole zeigt: Programm startet..., Kein Text vorhanden, Programm beendet.

5.4.4 Suche im Text

Ein 'gib aus' Block mit dem Text 'im Text' und dem Text 'das ist der Text.'. Dieser ist mit 'suche erstes Auftreten des Begriffs' und dem Text 'der' verbunden. Die Konsole zeigt: Programm startet..., 9, Programm beendet.

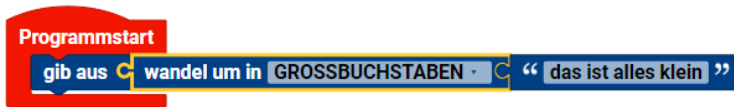
5.4.5 Extrahieren einzelner Zeichen

Ein 'gib aus' Block mit dem Text 'im Text' und dem Text 'das ist der Text.'. Dieser ist mit 'nimm' und der Zahl '6' verbunden. Die Konsole zeigt: Programm startet..., s, Programm beendet.

5.4.6 Extrahieren eines Textteils

Ein 'gib aus' Block mit dem Text 'im Text' und dem Text 'das ist der Text.'. Dieser ist mit 'nimm Teil ab', der Zahl '5', 'bis', der Zahl '7' und 'Buchstabe' verbunden. Die Konsole zeigt: Programm startet..., ist, Programm beendet.

5.4.7 Groß und Kleinschreibung ändern



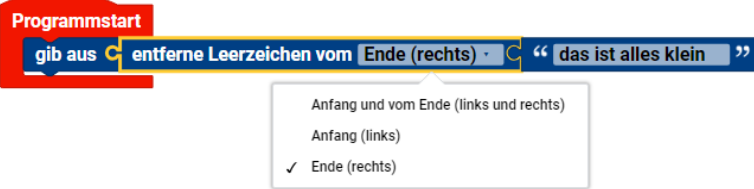
Konsole

```

Programm startet...
DAS IST ALLES KLEIN
Programm beendet.

```

5.4.8 Leerzeichen entfernen (vorn bzw. hinten)



Konsole

```

Programm startet...
das ist alles klein
Programm beendet.

```

5.4.9 Schriftgröße ändern

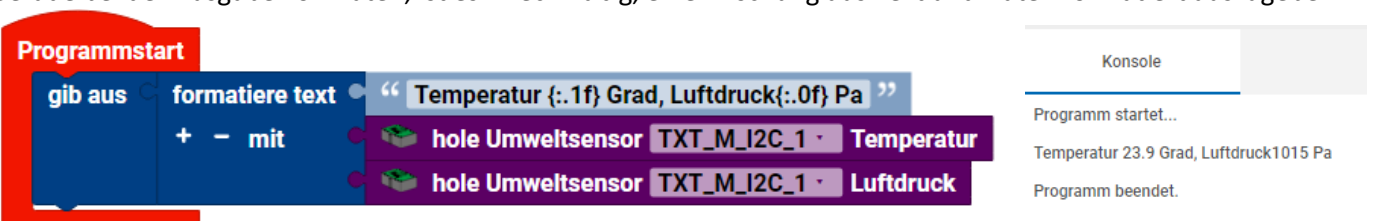
Wie bei der Syntax in HTML, können Texte formatiert ausgegeben werden. Die Formatierung kann entsprechend der Anwendung in der Anzeigekonfiguration oder im Programm bei der Textausgabe durchgeführt werden. Die Größe wird mit <h1> bis <h6> eingeleitet und mit </h1> bis </h6> abgeschlossen.

Weitere Formatierungen wären **fett** mit , *kursiv* mit <i> und unterstrichen mit <u>.



5.4.10 Texte formatieren

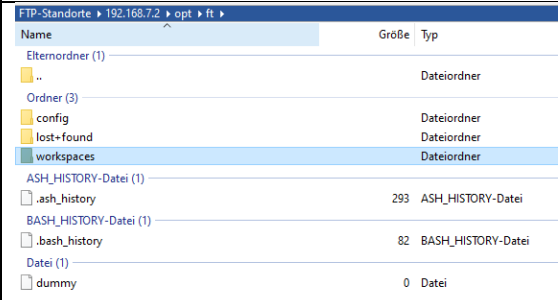
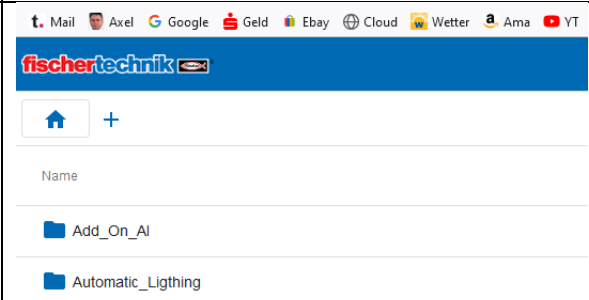
Gerade bei der Ausgabe von Daten, ist es zweckmäßig, eine Mischung aus Text und Daten formatiert auszugeben.



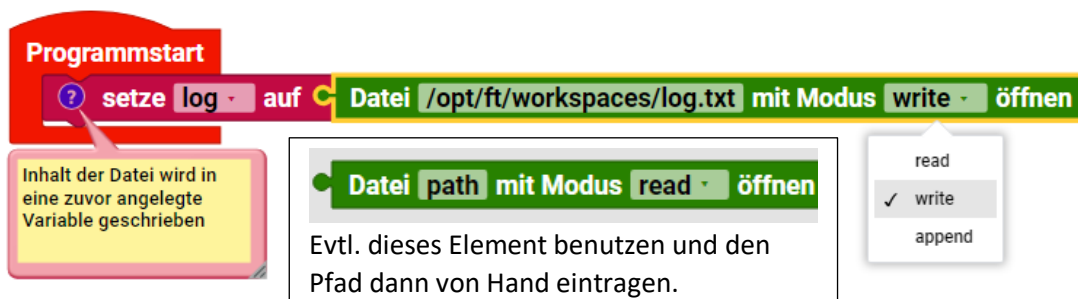
In jede geschweifte Klammer wird der folgende Wert aus dem Block "formatiere text" ausgegeben. Hierbei können auch die Anzahl der Nachkommastellen mit „:xf“ festgelegt werden. (x = Anzahl der Nachkommastellen)

5.5 Datei

Bei Arbeiten mit Dateien, werden diese nur auf dem TXT 4.0 gespeichert. Dabei ist der Pfad anfänglich vorgegeben: /opt/ft/workspaces/log.txt , um keine Systemdateien zu manipulieren.
Um die Dateien auslesen zu können, gibt es 2 Möglichkeiten:

Mit einem FTP-Programm Server: 192.168.7.2 (Bei USB) Port: 22 Benutzername: ft Passwort: fischertechnik	Mit dem Browser URL: 192.168.7.2 (Bei USB-Verbindung) Benutzername: ft Passwort: fischertechnik
	

5.5.1 Datei ... mit Modus... öffnen

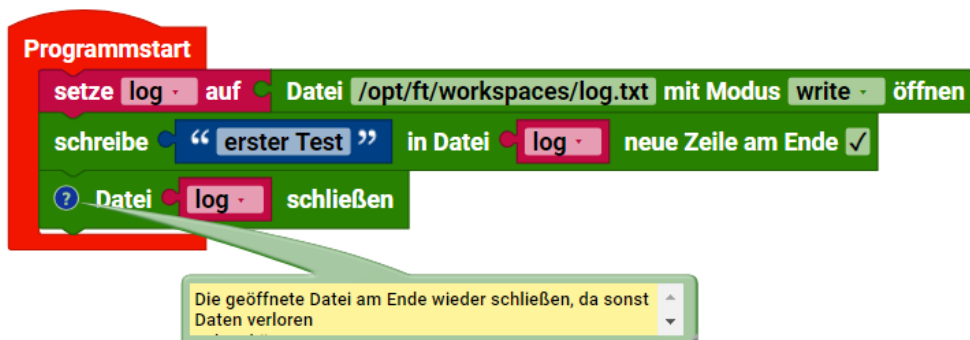


Sollte die Datei noch nicht existieren, wird sie hier automatisch angelegt.
Der Modus kann auf: lesen, schreiben oder anhängen gestellt werden.

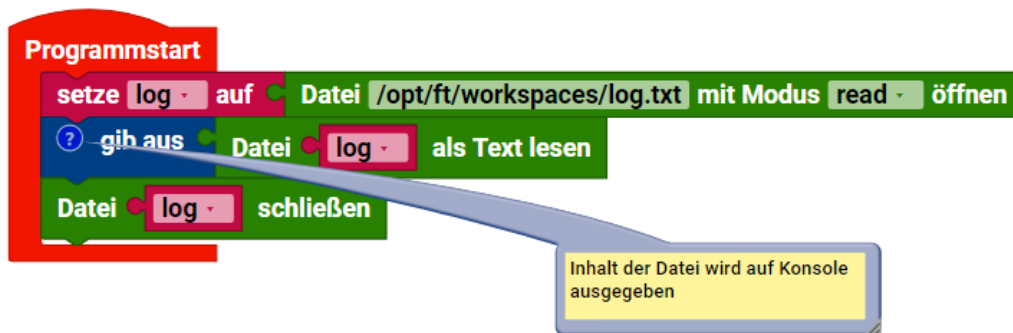
5.5.2 schreibe... in Datei... neue Zeile am Ende



5.5.3 Datei ... schließen



5.5.4 Datei... als Text lesen



Konsole

Programm startet...

erster Test

1
2

Programm beendet.

5.5.5 Liste... zeilenweise in die Datei ... schreiben

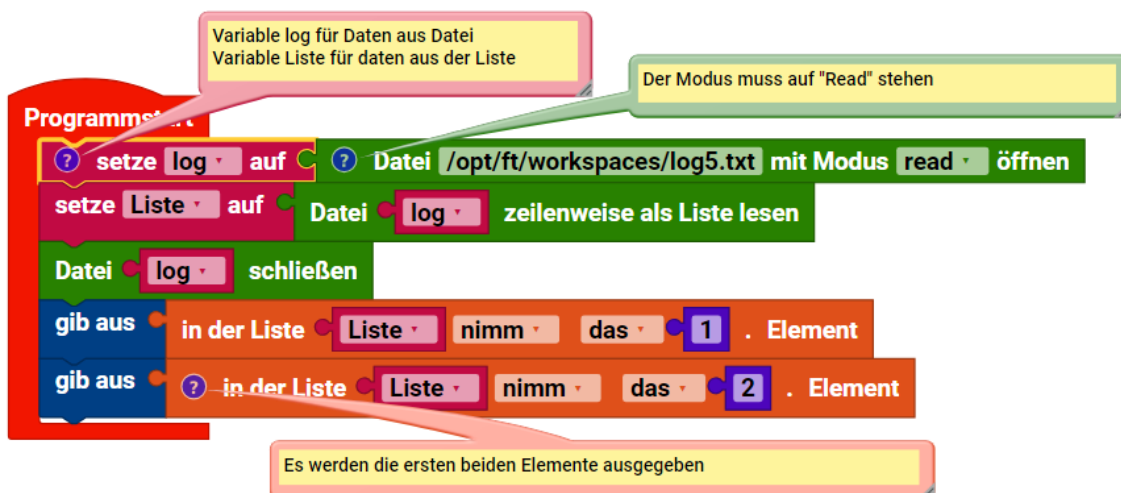


log4.txt - Editor

Datei Bearbeiten Format

1
2

5.5.6 Datei ... zeilenweise als Liste lesen



log5.txt - Editor

Datei Bearbeiten Format

Axel
Chobe
Testlog

Konsole

Programm startet...

Axel
Chobe

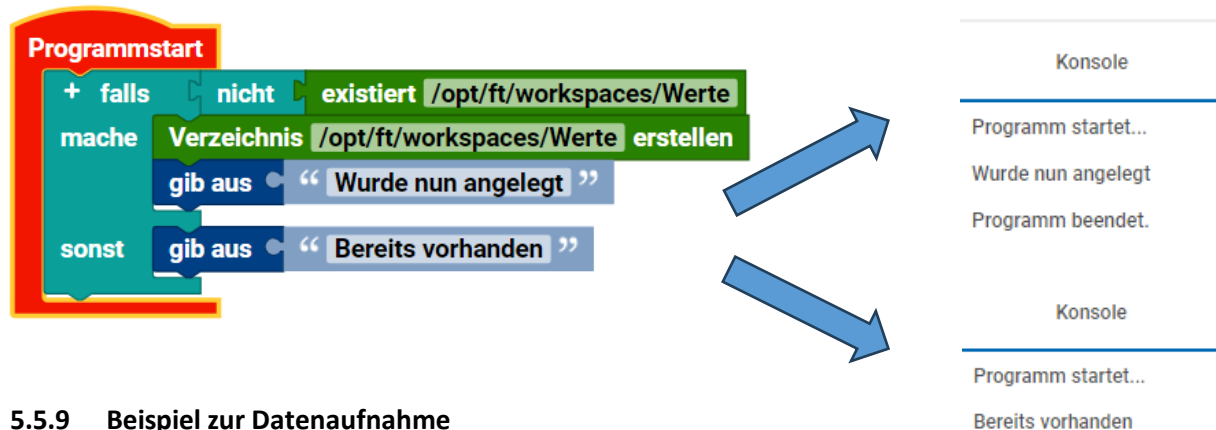
Programm beendet.

5.5.7 Verzeichnis path erstellen

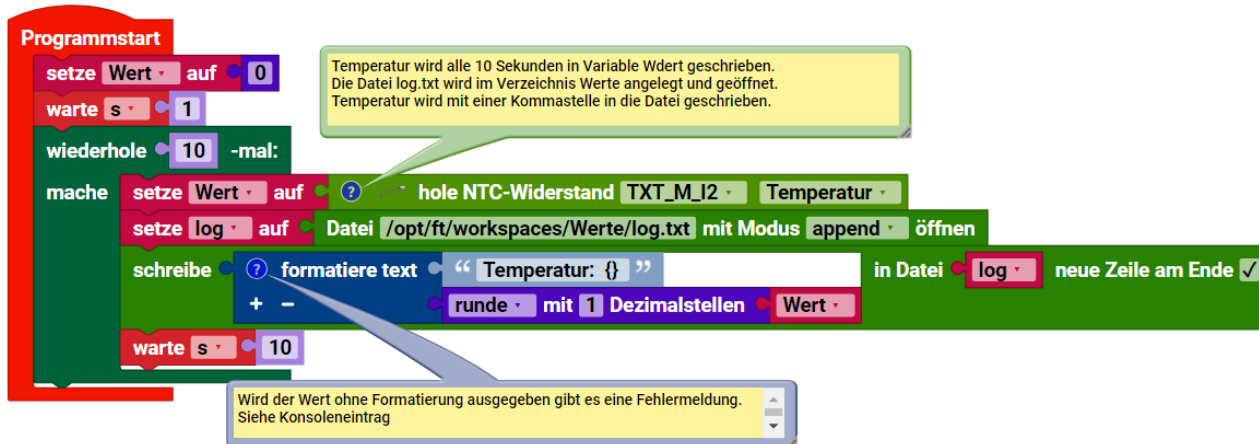


- Tachometer_and_Odometer_Thread
- Taxi_Meter
- Temp
- test_battery
- test_combi_sensor

5.5.8 existiert ...



5.5.9 Beispiel zur Datenaufnahme



*log.txt - Editor

Datei	Bearbeiten	Format	Ans
Temperatur:	25.1		
Temperatur:	25.1		
Temperatur:	24.7		
Temperatur:	25.1		
Temperatur:	24.7		
Temperatur:	24.8		
Temperatur:	25.4		
Temperatur:	25.0		
Temperatur:	25.3		
Temperatur:	25.0		

Konsole

Programm startet...

TypeError: unsupported operand type(s) for +: 'float' and 'str'

Programm beendet.

5.6 Datenstrukturen (Listen)

5.6.1 Liste erzeugen

Listen können aus Text, Zahlen oder Farben bestehen

Programmstart
setze Liste auf ? + - erzeuge Liste mit "eins"
"zwei"
"drei"

Farbelemente aus der Gruppe Util

Programmstart
setze Liste auf ? + - erzeuge Liste mit [Farbelemente]

Es gibt nun 3 mal ein Element mit dem Namen Test.

Programmstart
setze Liste auf ? erzeuge Liste mit 3 mal dem Element "Test"
gib aus in der Liste Liste nimm das 1. Element
gib aus in der Liste Liste nimm das 2. Element
gib aus in der Liste Liste nimm das 3. Element

Liste wird aus dem Text erstellt. Das Trennzeichen wird am Ende festgelegt.

Programmstart
setze Liste auf ? Liste aus Text erstellen "One,Two,Three" mit Trennzeichen ","
gib aus in der Liste Liste nimm das 1. Element
gib aus in der Liste Liste nimm das 2. Element
gib aus in der Liste Liste nimm das 3. Element

Konsole

Programm startet...

Test
Test
Test

Programm beendet.

Konsole

Programm startet...

One
Two
Three

Programm beendet.

5.6.2 Länge einer Liste ermitteln

Programmstart
setze Liste auf + - erzeuge Liste mit "eins"
gib aus Länge von Liste

Konsole

Programm startet...

1

Programm beendet.

5.6.3 Suche und manipulieren von Elementen in einer Liste

Gibt die Nummer des Elements aus auf den die Suche zutrifft.

Programmstart
setze Liste auf ? + - erzeuge Liste mit "eins"
"zwei"
"drei"
gib aus in der Liste Liste suche erstes Auftreten von "drei"

✓ erstes
letztes

Gibt die Nummer des Elements aus auf den die Suche zutrifft. Alternativ ist es hier auch möglich, ein Element zu löschen.

Programmstart
setze Liste auf ? + - erzeuge Liste mit "eins"
"zwei"
"drei"
gib aus in der Liste Liste nimm das 2. Element

✓ das
von hinten das
Erste
Letzte
Zufällig

✓ nimm
nimm und entferne
entferne

Konsole

Programm startet...

3

Programm beendet.

Konsole

Programm startet...

zwei

Programm beendet.

Ersetzt das zweite Element von zwei auf vier. Alternativ ist es auch möglich, ein Element einzufügen.

Programmstart

setze Liste auf + erzeuge Liste mit "eins", "zwei", "drei"

in der Liste Liste setze für das ein "vier"

gib aus in der Liste

gib aus in der Liste

gib aus in der Liste Liste nimm das 3. Element

✓ das
von hinten das
Erste
Letzte
Zufällig

Konsole

Programm startet...

eins
vier
drei

Programm beendet.

5.6.4 Sublist erstellen

Übergibt Elemente in eine neue Liste.

Programmstart

setze Liste auf + erzeuge Liste mit "eins", "zwei", "drei"

setze Teilliste auf in der Liste Liste nimm Teilliste ab erstes bis 2. Element

gib aus in der Liste Teilliste nimm das 1. Element

gib aus in der Liste Teilliste nimm das 2. Element

nimm Teilliste ab
nimm Teilliste ab von hinten
✓ nimm Teilliste ab erstes

✓ bis
bis von hinten
bis letztes

Konsole

Programm startet...

eins
zwei

Programm beendet.

5.6.5 Liste sortieren

Werte der Liste werden nach Vorgabe sortiert und in neu Liste eingetragen

Programmstart

setze Liste auf + erzeuge Liste mit "6", "2", "4"

setze Sortlist auf alphabetisch aufsteigend Liste sortieren

gib aus in der Liste Sortlist nimm Element

gib aus in der Liste Sortlist nimm Element

gib aus in der Liste Sortlist nimm das 3. Element

numerisch
✓ alphabetisch
alphabetisch, Großschreibung ignorieren

✓ aufsteigend
absteigend

Konsole

Programm startet...

2
4
6

Programm beendet.

5.6.6 Liste drucken

Gibt die Liste in der Form aus, die in der Programmierung typisch sind

Programmstart

setze Liste auf + erzeuge Liste mit "eins", "zwei", "drei"

gib aus Liste

Konsole

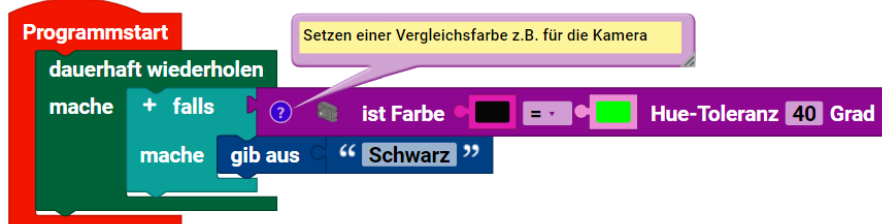
Programm startet...

['eins', 'zwei', 'drei']

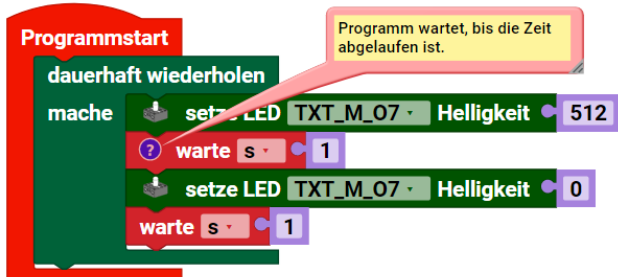
Programm beendet.

5.7 Util

5.7.1 Farbauswahl



5.7.2 Warte (Zeit)



5.7.3 Warte (Bedingung)



5.7.4 Thread

führe Funktion in einem Thread aus

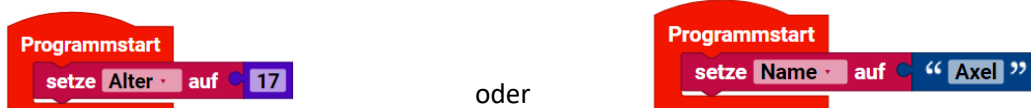
Siehe 5.9.4

5.8 Variablen

5.8.1 Erstellen



5.8.2 Festlegen

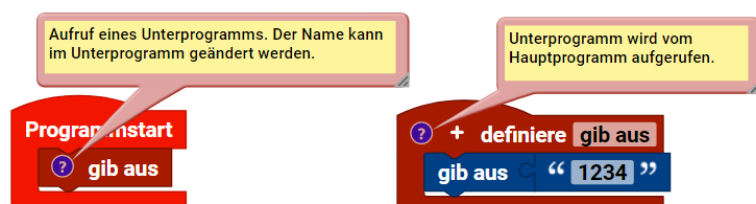


5.8.3 Wert ändern



5.9 Funktionen (Unterprogramme)

5.9.1 Einfacher Aufruf



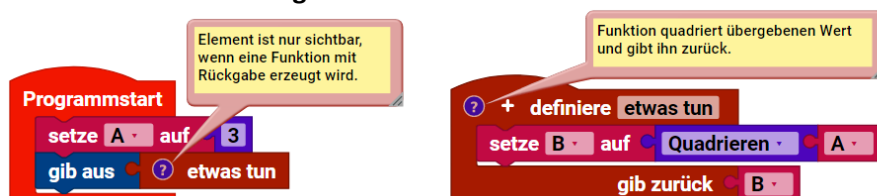
Konsole

Programm startet...

1234

Programm beendet.

5.9.2 Aufruf mit Rückgabe



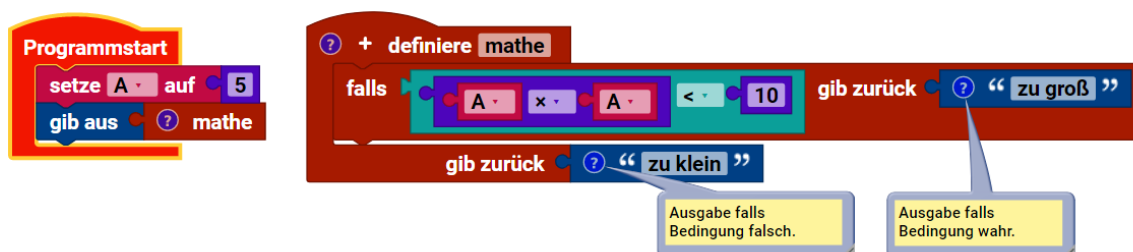
Konsole

Programm startet...

9.0

Programm beendet.

5.9.3 Aufruf mit Rückgabe und Bedingung



Konsole

Programm startet...

zu klein

Programm beendet.

5.9.4 Funktion als Thread ausführen

Im Normalfall werden Programme nacheinander ausgeführt. So wartet das Programm auch mit der Weiterführung, wenn ein Unterprogramm abgearbeitet wird. Da hier das Unterprogramm unendlich ist, läuft das Hauptprogramm nicht weiter.



Konsole

Satz2
Satz1
Satz2
Satz1
Satz2
Satz1

Deshalb wird hier das Unterprogramm 2 angewiesen, als Thread (also parallel) zu laufen.



Konsole



Satz2
Satz1
Satz2
Satz1
Satz2
Satz1

5.10 Machine Learning


Noch nicht dokumentiert.

5.11 Importe

Um den Quellcode übersichtlicher zu gestalten, ist es möglich, zusätzliche Module(Dateien) anzulegen. Dort können

z.B. Funktionen (Unterprogramme) abgelegt werden. Mit  wird das folgende Fenster geöffnet. Durch die Auswahl Quellcode wird eine neue Datei angelegt und als Reiter im Kopf angezeigt. 

Hier kann das Unterprogramm geschrieben werden.

Im Hauptprogramm, wird die Funktion  „Blink“ aus gewählt.

Neue Datei



Quellcode



Kamera



Anzeige



Bedienfeld



txt / csv / json

Funktionen

Machine Learning

Importe

Kommunikation

Sprachsteuerung

Programmstart

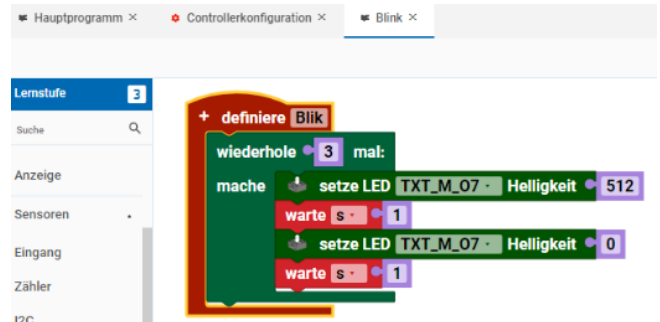
dauerhaft wiederholen

mache + falls

mache

Blink

ist Mini-Taster TXT_M_I3 geschlossen

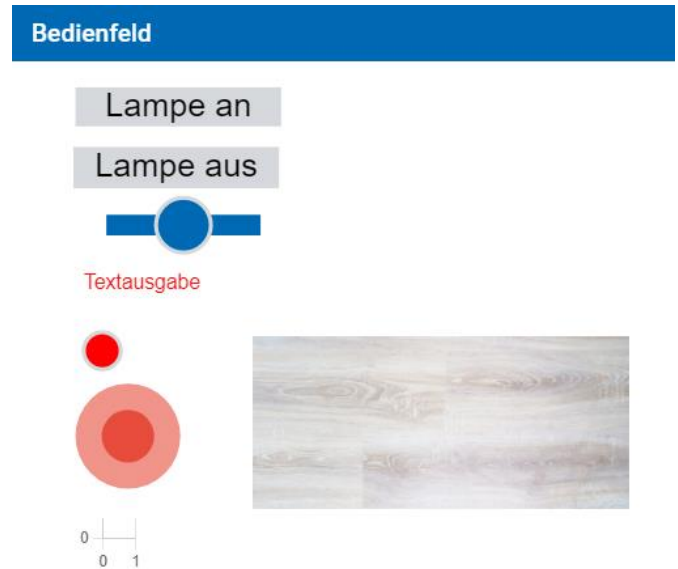
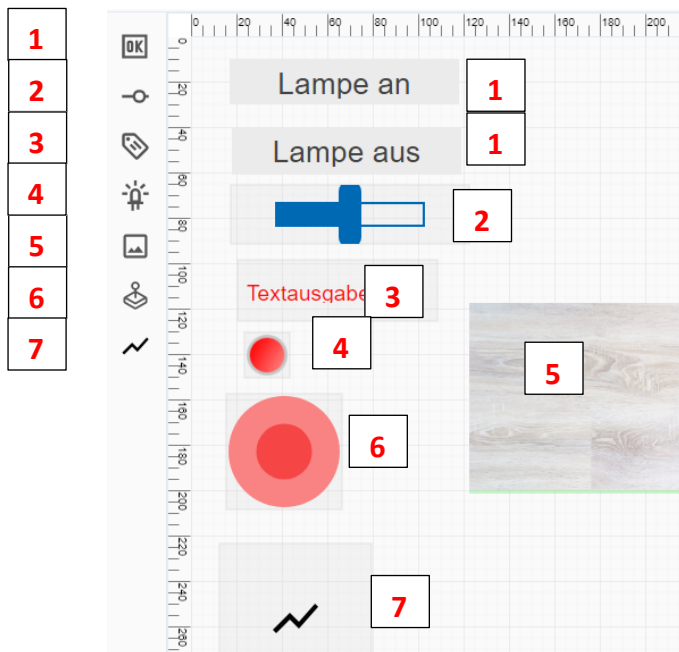


6 Kommunikation

6.1 Fernbedienung

Über eine Benutzeroberfläche, können Fahrzeuge oder Modelle ferngesteuert werden.

Bevor Befehle ausgeführt werden können, dauert es derzeit bis zu 20 Sekunden bis Aktionen möglich sind.



6.1.1 remote_button

Identität

Name :

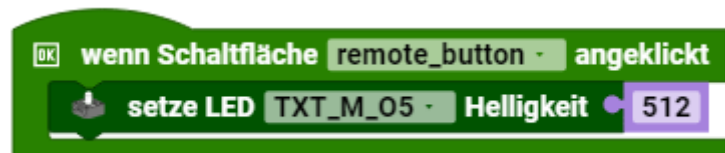
RCSButton

Text :

Schriftgröße : H1 H2 H3 T

Schriftstil : /

Durch Anklicken der Schaltfläche wird eine Aktion ausgeführt
Schriftgröße und Schriftstil können im Inspektor festgelegt werden.



6.1.2 remote_slider

Identität

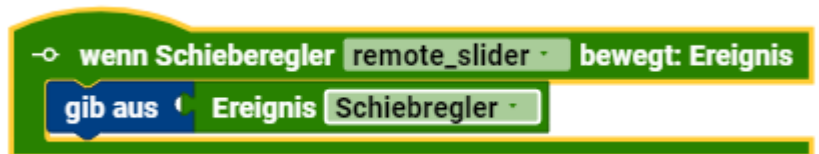
Name :

RCSlider

Ausrichtung: ☒ horizontal ☐ vertical

Von :
 Wert :
 Bis :

Die Bewegung des Sliders kann zur Steuerung von Motoren genutzt werden. Vorher werden der Wertebereich und der Anfangswert festgelegt. (Hier Ausgabe auf Konsole)



6.1.3 remote_label

Identität

Name :

RCLabel

Text :

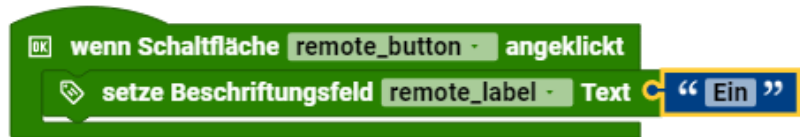
Schriftgröße : H1 H2 H3 T

Schriftstil : /

Ausrichtung :

Farbe :

Einerseits kann hier eine Beschriftung der Fernbedienung erfolgen. Andererseits können auch Rückmeldungen als Text erfolgen. (Siehe Beispiel). Zusätzlich zur Textgröße und Schriftstil, kann auch die Ausrichtung und Farbe im Inspektor voreingestellt werden.



6.1.4 remote_status_indicator

Identität

Name :

RCIndicator

Aktiv : ☒

Farbe :

Wie bei dem remote_label kann hier die Rückmeldung einer Aktion angezeigt werden. Oder man nutzt die Anzeige für Fehlermeldungen oder das Ende einer Aktion.



6.1.5 remote_image

Identität

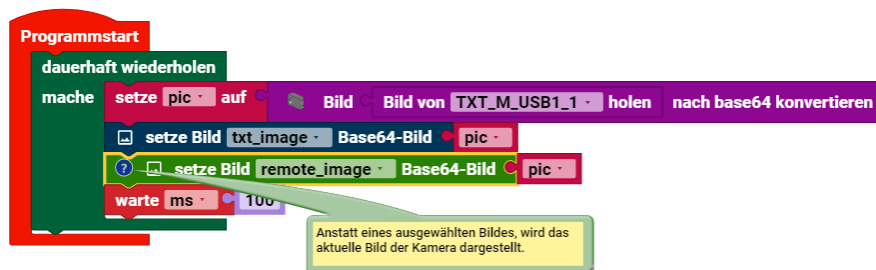
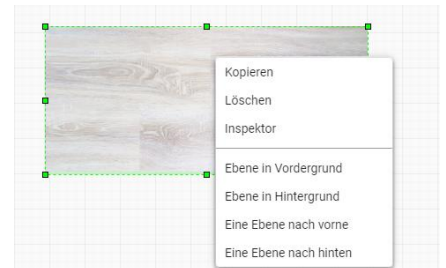
Name :

RCImage

Bild :

Hiermit lässt sich ein Bild auf die Bedienoberfläche legen.

Entweder als Detailbild zur besseren Erkennung oder sogar als Hintergrundbild des ganzen Bedienfeldes. Über das Kontextmenü kann dann die Ebene Hintergrund ausgewählt werden.



Eine zweite Möglichkeit, ist die Darstellung des Kamerabildes. Abhängig von der Größe des Bildes, kann es sehr lange dauern, bis das Bild dargestellt wird!!

6.1.6 remote_joystick

Identität

Name :

RCJoystick

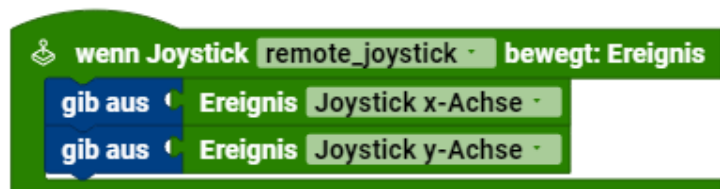
Von :

Bis :

Lock - X : ☐

Lock - Y : ☐

Die Bewegung des Joysticks kann in X und Y-Position ausgewertet werden. Im Inspektor können Anfangs und Endwert festgelegt werden. Im Beispiel erfolgt die Ausgabe auf der Konsole.



6.1.7 remote_chart

Größe

Breite: px

Höhe: px

Position

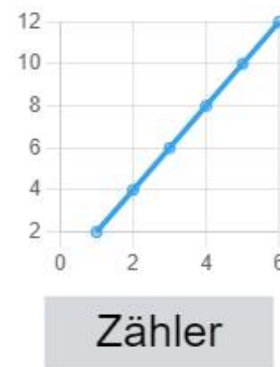
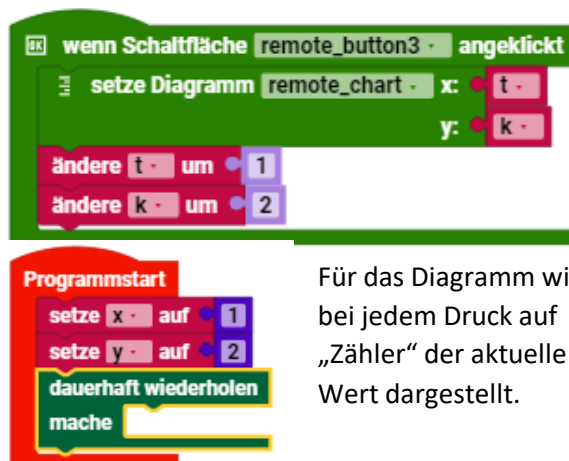
x: px

y: px

Identität

Name:

RCLineChart



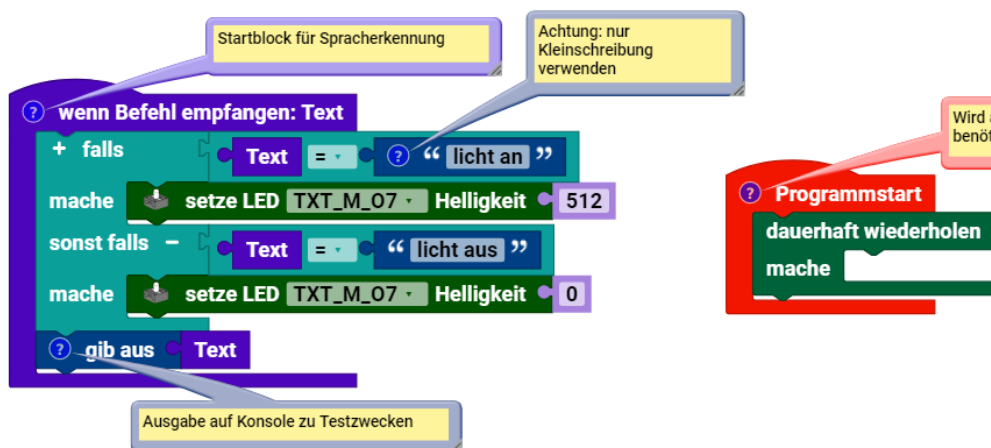
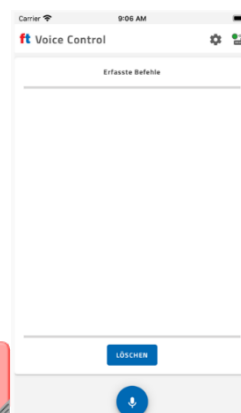
Für das Diagramm wird bei jedem Druck auf „Zähler“ der aktuelle Wert dargestellt.

6.2 Sprachsteuerung



Um die Sprachsteuerung zu nutzen, ist die App „Voice Control Fischertechnik“ auf dem Handy erforderlich. Anschließend meldet man sich auf dem TXT-Controller über WLAN an.

Der gesprochene Befehl wird wie im Beispiel ausgewertet und es erfolgt die entsprechende Aktion.



Konsole

Programm startet...

licht an
licht aus

6.3 Cloud / MQTT

Die Cloud ist unter „https://www.fischertechnik-cloud.com“ zu erreichen. Bevor man mit diesem Dashboard arbeiten kann, muss eine Anmeldung erfolgen.



Anschließend wird der Controller mit der Cloud gekoppelt werden.

Einstellungen -
Controller
hinzufügen

Einstellungen -
fischertechnik
Cloud

Controller registrieren

Name des Controllers *
TXT 4.0

Kopplungscode *

ABBRECHEN SPEICHERN

Bei fischertechnik registrieren

Nutzername

Passwort

Passwort-Wiederholung

Geburtsdag

E-Mail

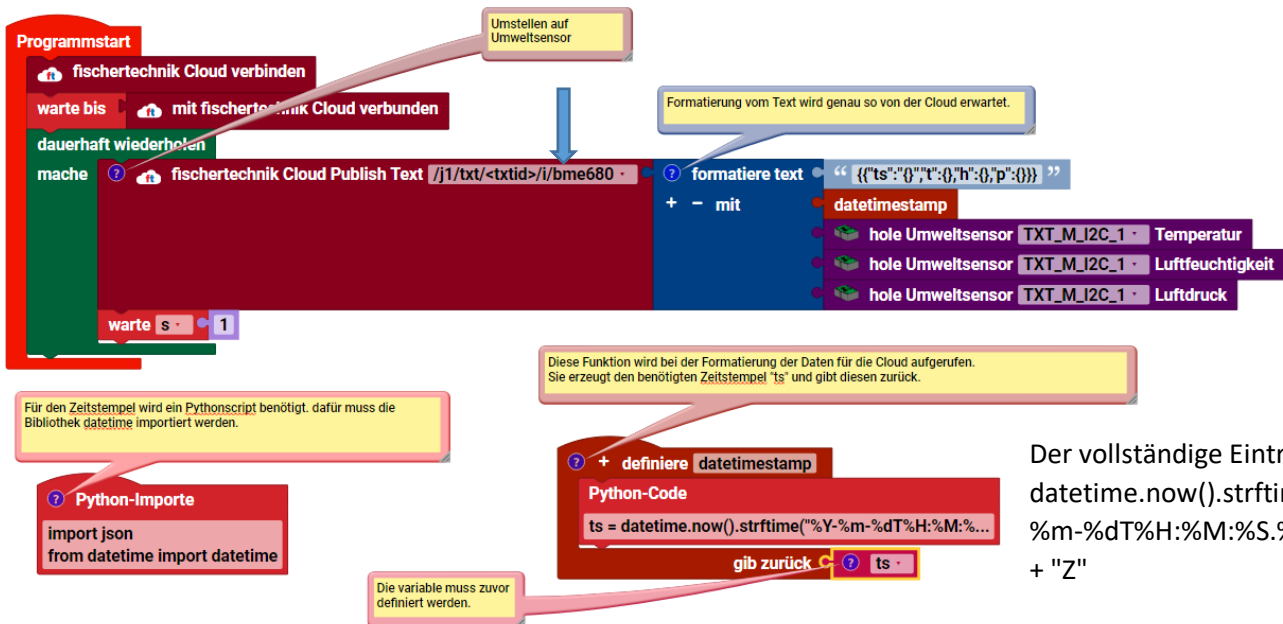
Eltern E-Mail

☐ Durch die Registrierung akzeptiere ich die Datenschutzerklärung.

☐ Ich bin kein Roboter

Registrieren

Du hast einen Account? Anmelden



Der vollständige Eintrag: ts= datetime.now().strftime("%Y-%m-%dT%H:%M:%S.%f")[:-3] + "Z"

Wenn alles klappt, sollten die 3 Werte (Temperatur, Luftfeuchtigkeit und Luftdruck) in den jeweiligen Fenstern der Cloud im Sekundentakt angezeigt werden.

