



Übersicht der RoboPro Befehle für Robo Interface

Zusammengestellt von Axel Chobe

09.03.2013



Ergänzung für Fischertechnik TXT-Controller

24.01.2017

Ergänzung für Fischertechnik TX-Controller

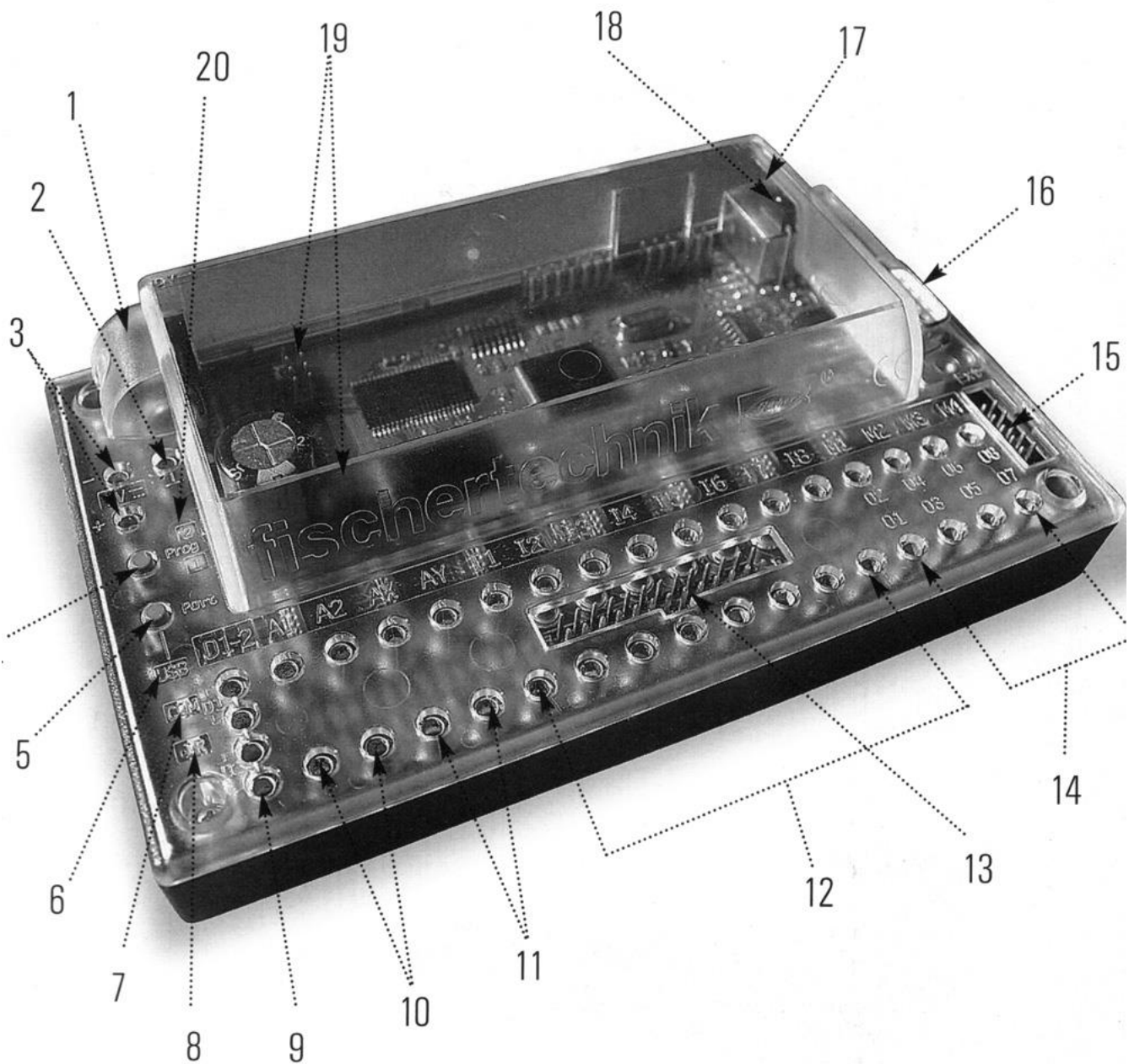
09.02.2018

Inhaltsverzeichnis:

1 Interface	4
2 Werkzeugleiste.....	6
2.1 Dateibearbeitung	6
2.3 Unterprogramme	6
2.4 Programm starten	6
2.5 Programme auf Interface laden	6
2.6 Gerät und Schnittstellen	6
1.7 Interface-Test.....	6
2.8 Debugger.....	7
3 Elementefenster.....	7
4 Grundelemente	8
4.1 Eigenschaften von Programmelementen bearbeiten.....	9
5 Unterprogramme	10
5.1 Unterprogramm erstellen	10
5.2 Unterprogramm einbinden.....	10
5.3 Unterprogramme kopieren.....	10
5.4 Befehlsübergabe aus Unterprogramm	11
5.5 Unterprogrammbibliotheken.....	11
5.6 Eigene Bibliothek	11
5.7 Bearbeiten von Unterprogrammssymbolen.....	11
6 Variablen	12
7 Befehle	14
8 Mehrere Prozesse	15
9 Erweiterungsmodule.....	16
10 Ändern der Interface-Seriennummer	16
11 Verzweigungen, Warten	17
12 Ein- und Ausgänge.....	18
13 Operatoren.....	20
14 Bedienfelder.....	21
16 Programme auf RoboPro laden und starten.....	22
15 Robo-Interface LED Error Codes	23

I TXT-Controller.....	24
I.I Anschlüsse	24
I.II Anschluß von Sensoren	25
I.III Universaleingänge umschalten	25
I.IV Weitere Eingangsmöglichkeiten.....	25
II.Display	26
II.I Größenanpassung.....	26
II.II. Steuerung und Anzeige über Display.....	26
II.III Textausgabe.....	27
III.Encodermotor.....	28
III.I Steuerung	28
III.II Zählengänge (Analog und Digital).....	28
III.III Steuerung mit Fernbedienung Control Set.....	29
III.IV Steuerung mit Fernbedienung Handy	31
IV Kamera	32
IV.I Bewegung.....	32
IV.II Farberkennung	33
IV.III Linienerkennung.....	33
IV.IV Ballerkennung	34
IV.V. Aussparung.....	34
V Soundausgabe	35
VI Oszillograph (Scope).....	35
VII Scratch.....	36
VIII Display und Taster TX-Controller	37
IX Zugriff auf Datenstruktur des TXT mit FTP	38
X Änderung der Sounddateien	40

1 Interface



Das ROBO Interface ermöglicht die Verständigung zwischen dem PC und einem Modell. Es dient dazu, die Befehle der Software so umzuwandeln, dass z.B. Motoren angesteuert und Signale von Sensoren verarbeitet werden können.

Die Stromversorgung erfolgt mit 9V Gleichstrom mit mindestens 1000 mA. Stromaufnahme 50 mA.

Der Prozessor ist ein M30245, 16 Bit, Taktfrequenz 16MHz, 128 kByte RAM, 128 kByte Flash .

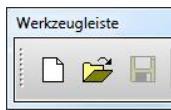
1	DC Buchse für Netzgerät
2	Massebuchse
3	Anschluß für Akkuset
4	Taste zur Auswahl der gespeicherten Programme (entsprechende LED leuchtet Prog 1/Prog2) kurzes drücken startet das Programm zum Starten von Programmen aus dem Flash-Speicher müssen beide LED leuchten
5	Taste zum Festlegen der Schnittstelle; es leuchtet dann die LED, die der Schnittstelle zugeordnet ist
6	LED für Schnittstelle für USB
7	LED für Schnittstelle Seriell
8	LED für Infrarot (IR); durch mehrfaches drücken der Taste 5 bis LED 8 leuchtet, können die Ausgänge ohne Anschluß am PC direkt über den Handsender bedient werden. Ist diese Funktion aktiv, sind die anderen Schnittstellen abgeschaltet
9	Eingänge für Abstandssensor D1 und D2
10	Analoge Spannungseingänge A1 und A2 zum Anschluß von analogen Sensoren die eine Spannung von 0 bis 10 Volt abgeben (Auflösung 10 Bit)
11	Analoge Widerstandseingänge AX und AY zum Anschluß von Potentiometer, Foto- oder NTC-Widerständen. Messbereich 0 – 5,5 KOhm; Auflösung 10 Bit
12	Digitale Eingänge I1 bis I8 für digitale Sensoren; Spannungsbereich 9 V, Schaltschwelle für Ein/Aus bei ca. 2,6 V; Eingangswiderstand ca. 10 KOhm
13	26-polige Stiftleiste zum herausführen der Ein- und Ausgänge über Flachbandkabel
14	Anschluß von 4 Motoren (vorwärts, rückwärts, aus, 8 Geschwindigkeitsstufen) mit 9 V – Alternativ können 8 Lampen angeschlossen werden. (zweiter Pol an Massebuchse 2)
15	Erweiterungsstecker für I/O-Extension; dadurch kann die Anzahl der Ein- und Ausgänge erweitert werden (besitzt 4 Motorausgänge, 8 digitale Eingänge und einen analogen Widerstandseingang)
16	serielle Schnittstelle
17	USB Schnittstelle
18	Infrarot (IR)-Eingang für Handsender
19	Erweiterungsstecker für ROBO RF Data Link; damit ist kein Verbindungskabel zwischen PC und Interface mehr erforderlich; PC-seitig wird der RF Data Link über USB angeschlossen (2,4 GHz) Reichweite 10 Meter
20	Die rote LED zeigt Fehler im Interface an Dauerleuchten – Versorgungsspannung außerhalb der Nennspannung (Interface schaltet ab) Blinkt die LED, hat der Prozess einen Fehler entdeckt; der Fehlercode kann über die restlichen grünen LED's erkannt werden (siehe letztes Kapitel Error-Codes)



Optionales Zubehör für Robo Pro

2 Werkzeugleiste

2.1 Dateibearbeitung



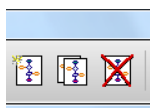
- Ein neues ROBO Programm anlegen
- Ein bestehendes ROBO Programm öffnen
- Das aktuelle Programm in einer Datei speichern

2.2 Elemente löschen



Werkzeug anklicken und auf dem entsprechenden Element mit linker Maustaste löschen

2.3 Unterprogramme



- Neues Unterprogramm erzeugen
- Aktuelles Unterprogramm kopieren
- Aktuelles Unterprogramm löschen

2.4 Programm starten

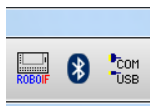


- Programm im Online-modus starten
- Alle laufenden Programme anhalten

2.5 Programme auf Interface laden



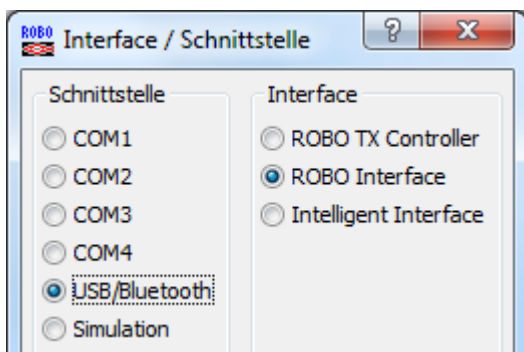
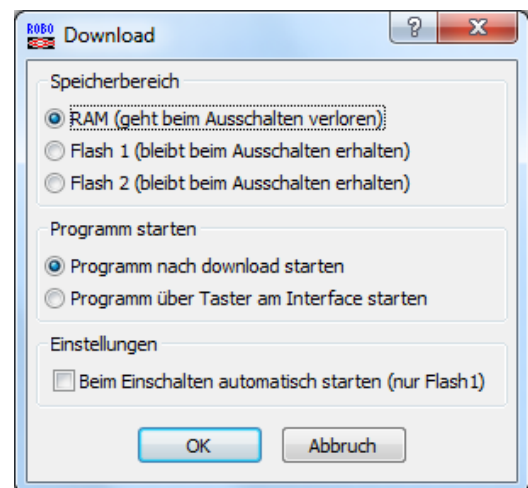
- Mit dem Werkzeug wird nebenstehendes Fenster geöffnet.
- Es gibt einen flüchtigen (RAM) und zwei feste Speicherbereiche. (Siehe Kapitel 16)



2.6 Gerät und Schnittstellen

Auswahl zwischen Interface oder TX

Bluetooth nur für TX

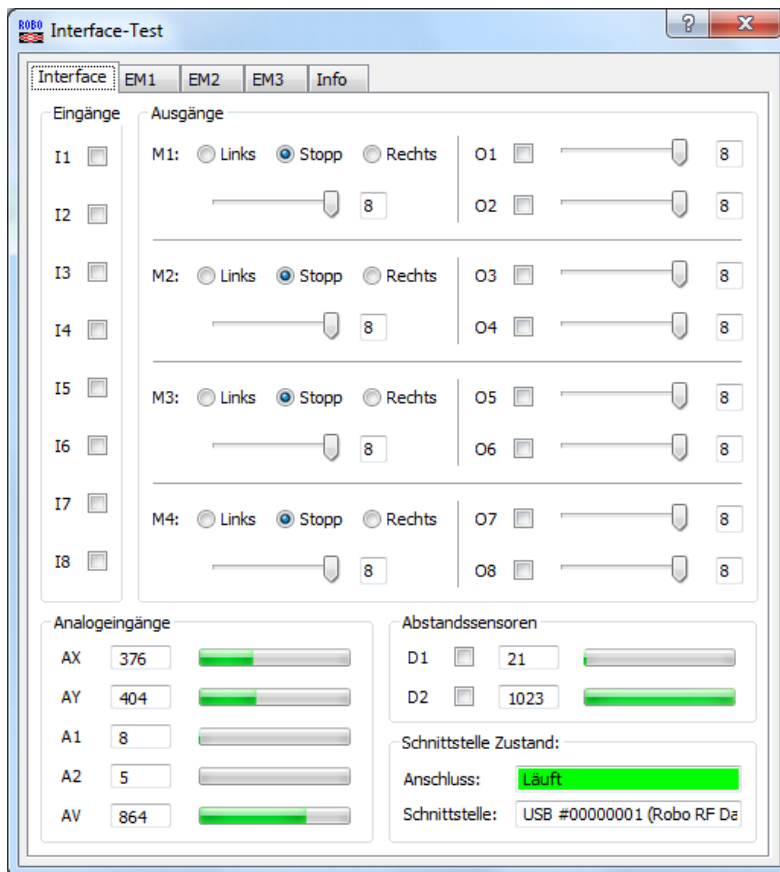


Entsprechend Interface und Schnittstelle auswählen:
Hinweis: Beim ROBO RF Data Link wird die USB-Schnittstelle gewählt

1.7 Interface-Test



- Der Aufruf erfolgt über den Knopf „Interface testen“ in der Werkzeugleiste.
- Es werden alle vorhandenen Ein- und Ausgänge des Interface angezeigt.
- Voraussetzung für die Funktionalität ist der Schnittstellen Zustand. (Läuft)



Digitaleingänge I1 – I8:

Die angeschlossenen Sensoren können nur die Zustände „0“ und „1“ annehmen. Dabei kann es sich um Schalter, Fototransistoren oder Reedkontakte handeln.

Motorausgänge M1 – M4:

Aktoren z.B. Motor lassen sich in der Geschwindigkeit und in der Richtung steuern.

Lampenausgänge O1 - O8:

Alternativ können auch 8 Lampen angeschlossen werden. Der zweite Anschluss wird dabei an Masse gelegt.

Analogeingänge AX – AY:

Zum messen des Widerstandes der angeschlossenen Sensoren wie NTC, Potentiometer oder Fotowiderstand.


Analogeingänge A1 – A2:

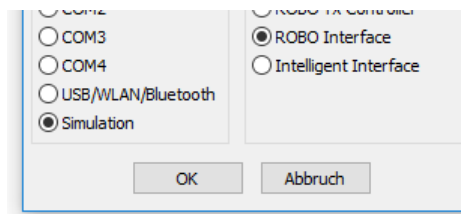
Die beiden Eingänge messen eine Spannung von 0 – 10 V.

Abstandssensoren D1 – D2:

Hier lassen sich nur spezielle Abstandssensoren anschließen

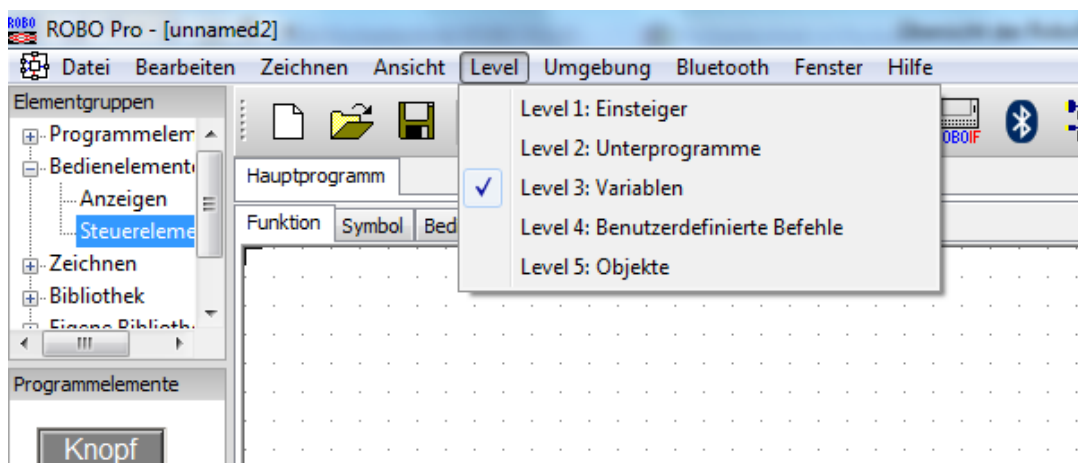
2.8 Debugger

- Interface Schnittstelle auf „Simulation“
- Fenster öffnen „Interface testen“ (Siehe 1.7)
- Debugmodus starten 
- Die Eingänge können jetzt im Testfenster betätigt werden. Lampen bzw. Motoren werden nun im Testfenster jeweils aktiv.



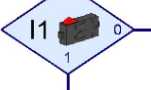
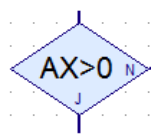

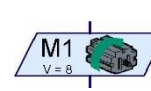


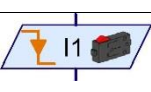
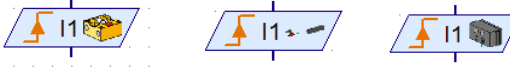
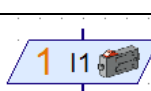


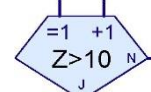
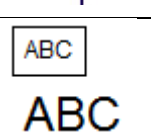


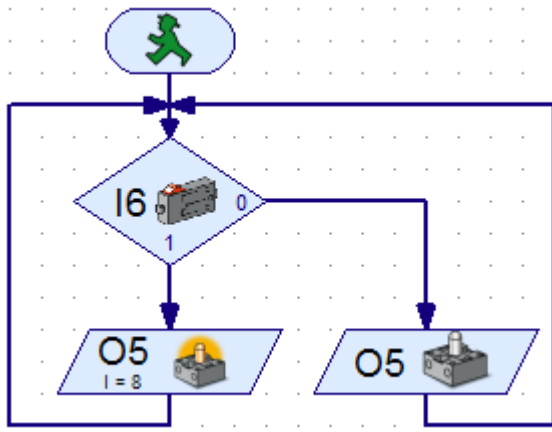
3 Elementfenster

Im Elementfenster (linke Seite) befinden sich die Programmelemente. Die Ansicht kann nach bestimmten Kriterien umgeschaltet werden. Dabei ist das entsprechende Level auszuwählen. Ab Level 2 teilt sich das Fenster in die Elementgruppen und die Programmelemente

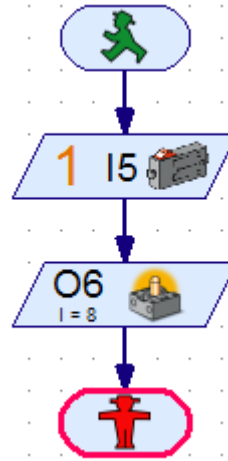


4 Grundelemente

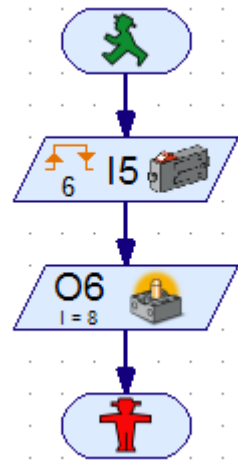
Symbol	Bezeichnung	Erläuterung
	Programm Beginn	Steht zu Beginn eines jeden Prozesses. Ein Programm kann mehrere Prozesse enthalten und quasi parallel abarbeiten.
	Programm Ende	Beendet einen Prozess. Soll der Prozess oder das Programm endlos laufen entfällt dieses Element.
	Verzweigung Digital	Je nach Zustand des digitalen Eingangs verzweigt das Programm in den entsprechenden Zweig. (Beispiel 1)
	Verzweigung Analog	Der Wert am Analogeingang wird mit dem eingestellten Wert in der Abfrage verglichen und dem entsprechend der 1 oder der 0-Zweig ausgeführt. Der Vergleichsoperator (<>=) kann entsprechend gewählt werden. Mit dem Sensortyp kann man den Sensor auf Widerstand, Spannung oder Abstand einstellen. (Beispiel 5)
	Wartezeit	Das Element Wartezeit verzögert den Programmablauf um die eingestellte Zeit. Die Zeit ist im Bereich von 1ms bis 500h einstellbar.
	Motorausgang Magnetventil Elektromagnet Summer, Lampe	Steuert Motor M1bis M4. Einstellbar ist Rechts, Links, Stop und die Geschwindigkeit. (Wert 1 – 8) 
	Lampenausgang	Schließt man den einen Lampenpol an Masse und den anderen an die Ausgänge 0 bis 8, lassen sich damit bis zu 8 Lampen(Relais, Magnete) ansteuern. (erst ab Level 2 sichtbar)
	Warten auf Eingang Taster	 Fototransistor, Reed-Kontakt, Spurensucher
	Warten auf Eingang geschlossen	Der Programmablauf wartet auf 1. d.h. der Eingang ist geschlossen. (Beispiel 2)
	Warten auf Eingang Wechsel von 1 auf 0	Der Programmablauf wartet auf eine Änderung von 1 auf 0 am angeschlossenen Sensor.
	Warten auf Eingang Wechsel 0-1 oder 1-0	Der Programmablauf wartet auf eine beliebige Änderung am angeschlossenen Sensor. Von 0 auf 1 oder von 1 auf 0.
	Impulszähler	Der Programmablauf wartet auf eine festgelegte Anzahl beliebiger Änderung am angeschlossenen Sensor. (Beispiel 3)
	Zählschleife	Mit der Zählschleife wird ein Programmteil mehrfach ausgeführt. Der =1 Eingang setzt den Zähler auf 1, der +1 Eingang erhöht den Zähler jeweils um 1. Wird der eingestellte Zählerwert erreicht, wird die Schleife am J-Ausgang verlassen. (Beispiel 4)
	Beschriftung	Zur Übersichtlichkeit einer Schaltung, kann es hilfreich sein eine Beschriftung hinzuzufügen. Es sind zwei Schriftgrößen möglich. (Beispiel 6)



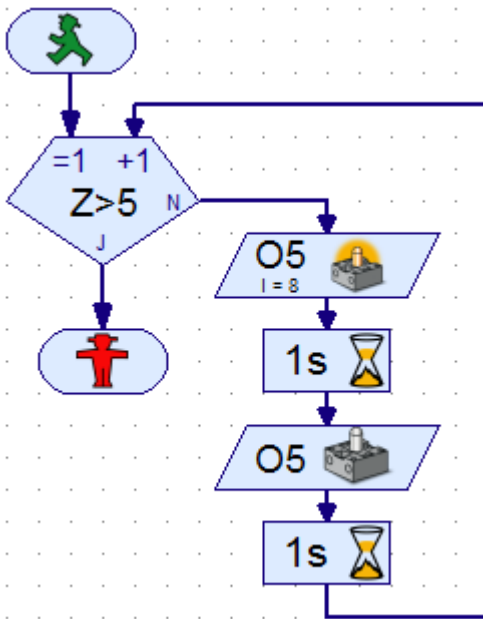
Beispiel 1



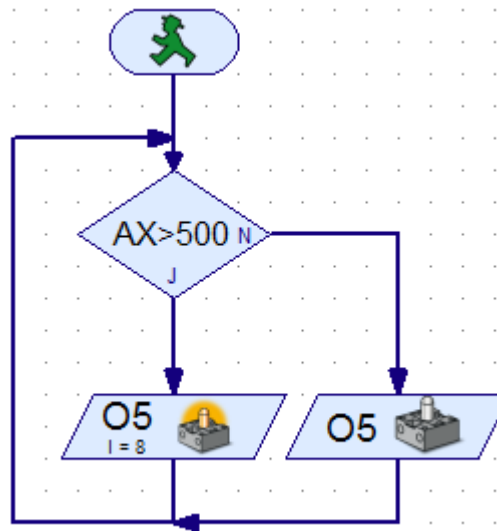
Beispiel 2



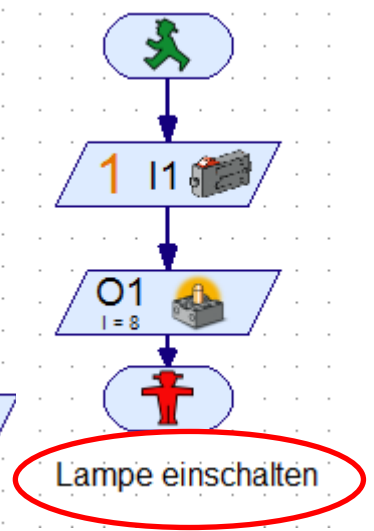
Beispiel 3



Beispiel 4 Lampe wird 5 mal geschaltet



Beispiel 5 AX = Fototransistor







Beispiel 6 Text

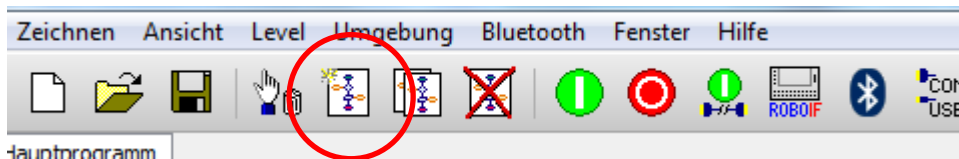
4.1 Eigenschaften von Programmelementen bearbeiten

Mit der rechten Maustaste kann ein Element angeklickt werden, um das Einstellfenster zu öffnen.

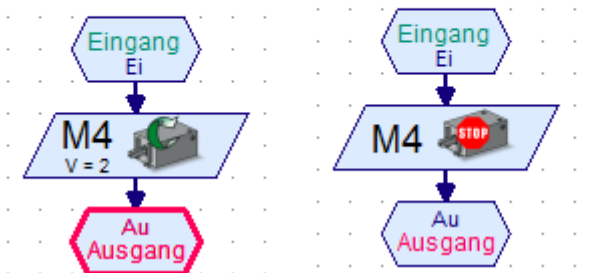
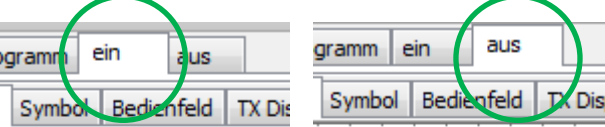
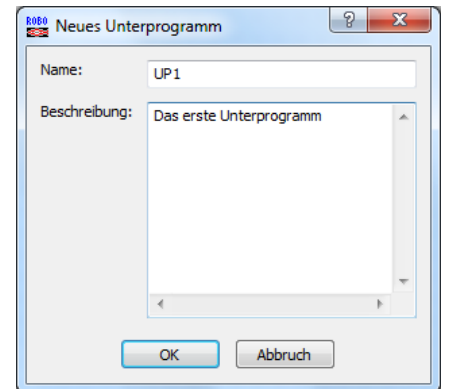
5 Unterprogramme

Symbol	Bezeichnung	Erläuterung
	Unterprogramm Eingang	Hier wird die Programmführung vom Hauptprogramm an das Unterprogramm übergeben
	Unterprogramm Ausgang	Hier wird die Programmführung vom Unterprogramm zurück an das Hauptprogramm gegeben
	Unterprogramm-Befehlseingang	Hier werden Befehle an das Unterprogramm übergeben.
	Unterprogramm-Befehlsausgang	Hier werden Befehle aus dem Unterprogramm nach außen übergeben.

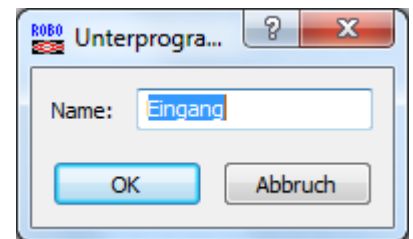
5.1 Unterprogramm erstellen



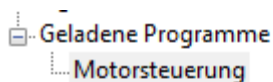
Nach Betätigung des Buttons öffnet sich ein Fenster für die Eingabe von Namen und Beschreibung des Unterprogramms.



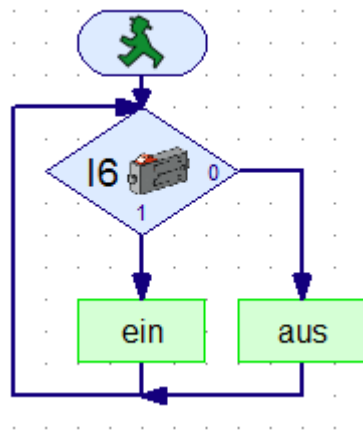
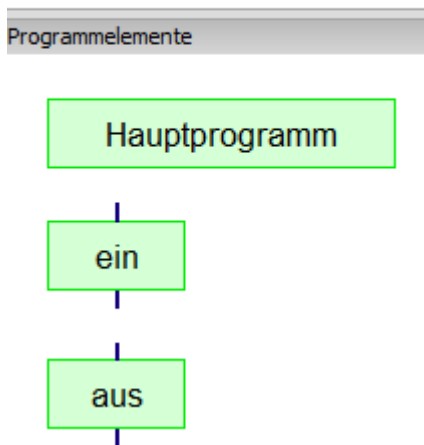
Neben dem Hauptprogramm gibt es nun neue Reiter mit dem entsprechenden Namen des Unterprogrammes. Die Ein- und Ausgangsbezeichnung kann über die rechte Maustaste umbenannt werden. Maustaste auf das Feld



5.2 Unterprogramm einbinden



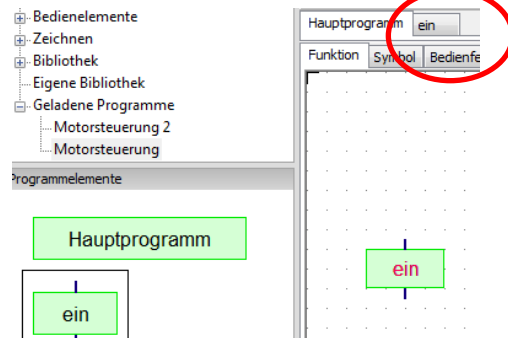
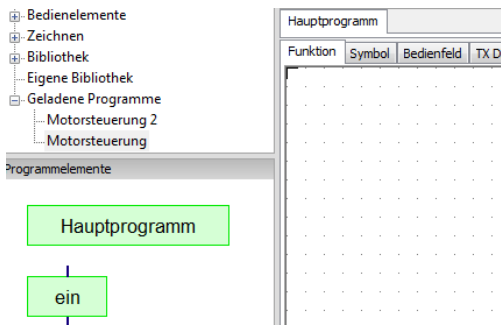
Nach dem speichern (hier Motorsteuerung) gibt es in der Elementegruppe „Geladene Programme“ die Unterprogramme als Programmelemente.



Diese Programmelemente werden nun im Hauptprogramm benutzt.

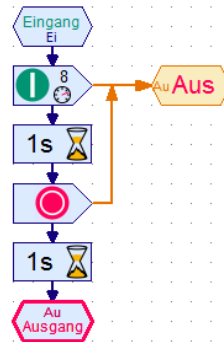
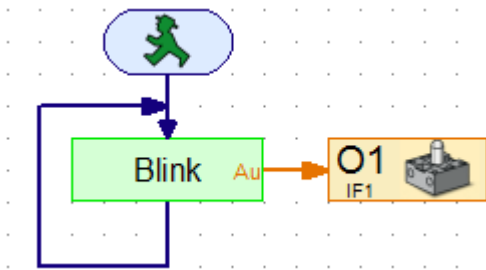
5.3 Unterprogramme kopieren

Unterprogramme können von einer Datei in eine andere kopiert werden. Dazu die beiden Dateien öffnen. Unter dem Menüpunkt Fenster, die Datei ohne Unterprogramme aufrufen.



Wird nun ein Unterprogramm in das rechte Fenster gezogen, gibt es sofort einen neuen Reiter mit dem Unterprogramm.

5.4 Befehlsübergabe aus Unterprogramm



Unterprogramm Blink

Dadurch können Unterprogramme für verschiedene Lampen oder Motoren eingesetzt werden

5.5 Unterprogrammbibliotheken

Es gibt drei Möglichkeiten auf Unterprogramme zuzugreifen:

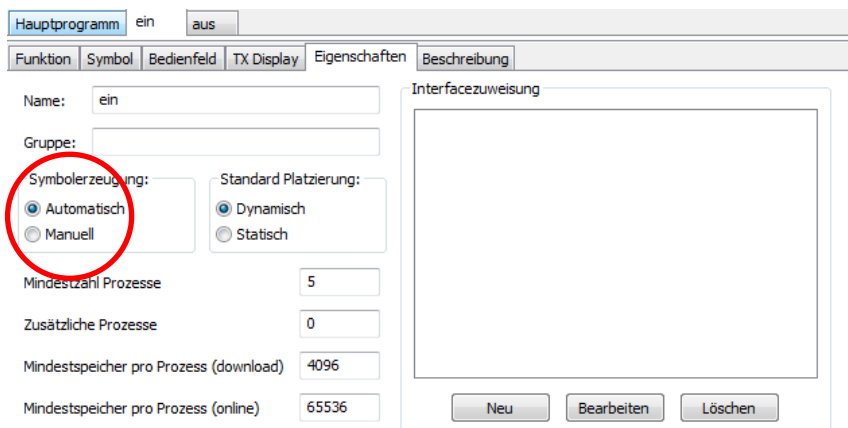
- Bibliothek: Vorgefertigte Unterprogramme für die entsprechenden Baukästen
- Eigene Bibliothek: Sammlung aller eigenen gespeicherten Unterprogramme
- Geladene Programme: Unterprogramme von aktuell geöffneten Dateien

5.6 Eigene Bibliothek

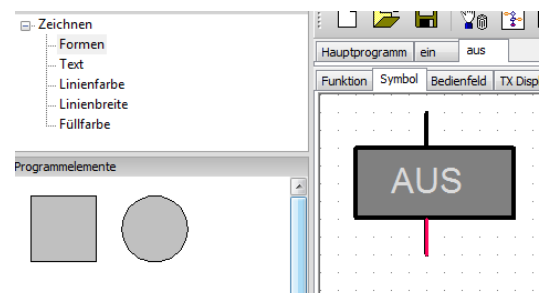
Vorraussetzung ist, das ein Verzeichnis angelegt wurde, indem die Dateien mit den Unterprogrammen gesammelt werden. Über den Menüpunkt „Datei/ Eigenes Bibliotheksverzeichnis“ wird dieser Ordner eingebunden. Die Unterprogramme können nun unter „Eigene Bibliothek“ und dem entsprechenden Dateinamen eingesetzt werden.

5.7 Bearbeiten von Unterprogrammssymbolen

Das vorgegebene Symbol kann verändert werden, um besser zu verdeutlichen, was das Programm macht.



Dazu den Reiter mit dem Unterprogramm und den Reiter Eigenschaften aufrufen. Die Symbolerzeugung von „Automatisch“ auf „Manuell“ umschalten.



In der Elementgruppe „Zeichen“ kann das Symbol verändert werden.

6 Variablen

Eine Variable ist ein Element, das einen Zahlenwert zwischen -32767 und 32767 speichern kann. Im Eigenschaftsfenster kann der Name und der Anfangswert der Variable geändert werden.

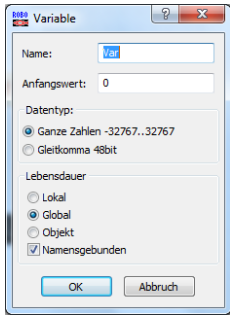
Globale Variablen

Es wird für Variablenelemente mit gleichem Namen nur eine globale Variable abgelegt. Sie können in verschiedenen Unterprogrammen vorkommen.

Lokale Variablen

Eine lokale Variable verhält sich fast genauso wie eine globale Variable, mit dem Unterschied, es gibt sie nur in dem Unterprogramm, indem sie definiert ist.

Eigenschaften



Unter Name kann der Name geändert werden

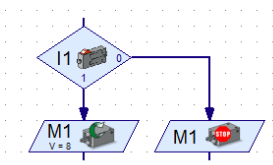
Unter Anfangswert kann der Wert eingegeben werden. Sie behält den Wert, bis sie über einen =, + oder – Befehl einen neuen Wert bekommt.

Variablentyp ist für Variablen in Unterprogrammen wichtig. Im Hauptprogramm haben beide den gleichen Effekt.

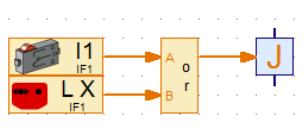
Symbole

Symbol	Bezeichnung	Erläuterung
	Variable(global)	Eine (globale) Variable, welche im gesamten Programm gültig ist. Es könne ganze oder Gleitkomma Zahlen verwendet werden.
	Lokale Variable	Eine lokale Variable, welche nur im jeweiligen Unterprogramm gültig ist. (Der Name und der Anfangswert können verändert werden- Kontext)
	Konstante	Konstanten beinhalten einen festen Wert.
	Befehl zuweisen (Beispiel Befehl Plus)	Erhöht den Wert des angeschlossenen Elementes um1. (Beispiel 1) (Der Befehlseingang kann abgeschaltet werden) (Beispiel 2)
	Timervariable	Die Timervariable zählt von einem eingestellten Wert bis auf Null runter. Mit ihr sind Zeitmessungen möglich. (Beispiel 3)
	Liste (Beispiel 4 und 5) s: set (Eingabe) r: read (Ausgabe) l: links Speicherplatz	Das Element Liste speichert mehrere Werte hintereinander ab. Bei der Definition kann die List auch schon mit Werten gefüllt werden. Eine Liste kann aus einer CSV-Datei gefüllt werden oder in diese schreiben. l: rechts Anzahl der belegten Speicherplätze
	Befehl Wert Anhängen -32767 bis 32767 (Beispiel 4)	Hängt den übermittelten Wert an die Liste an. Die Liste wird um ein Element vergrößert. (Der Dateneingang muss erst aktiviert werden)
	Befehl Entfernen aus Liste 0 bis 32767	Löscht eine einstellbare Anzahl von Werten vom Ende der Liste. Der mit dem Befehl übermittelte Wert ist die Anzahl der zu löschenden Elemente.
	Befehl Werte ver- tauschen 0 bis 32767	Vertauscht das angegebene Element mit dem Element am Listenbeginn. Der mit dem Befehl übermittelte Wert ist die Nummer des Elementes, das vertauscht werden soll.

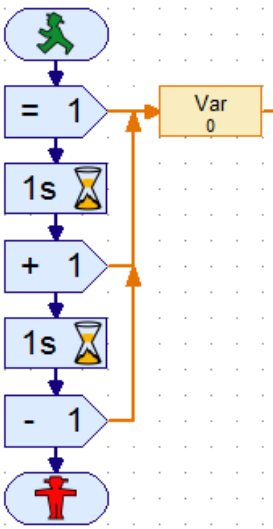
Hinweis:



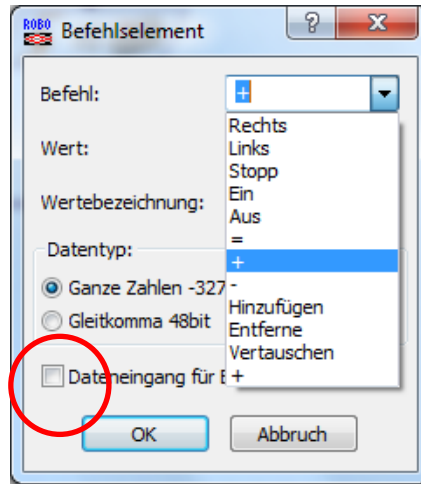
Beispiel für Kontroll-
Fluss (schwarze Linien)



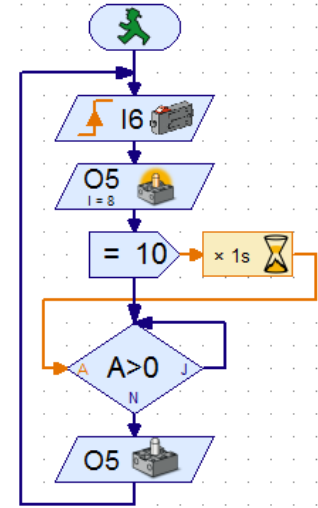
Beispiel für Datenfluss
(orange Linien)



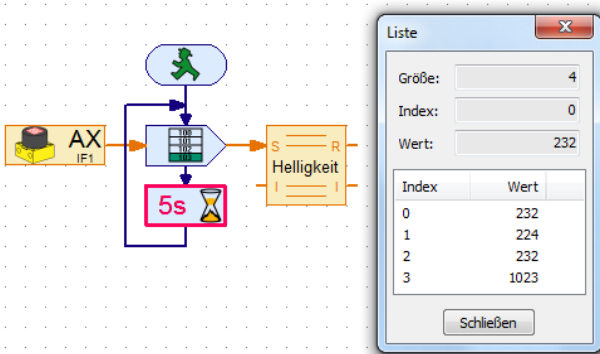
Beispiel 1



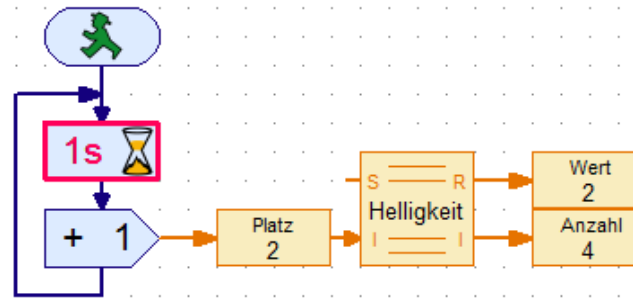
Beispiel 2



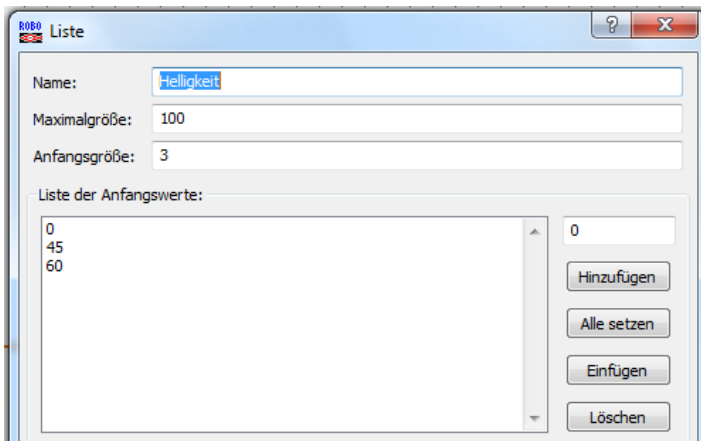
Beispiel 3



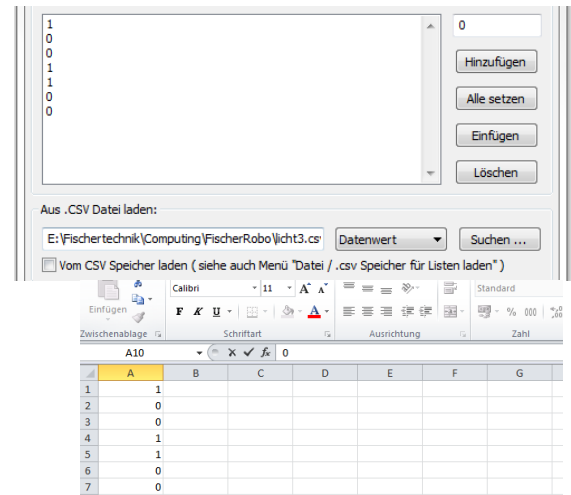
Beispiel 4 Werte einlesen (und anzeigen lassen)



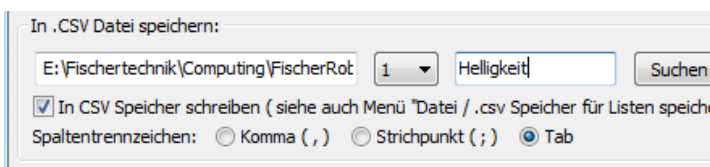
Beispiel 5 Werte auslesen



Beispiel 6 Werte manuell eingeben

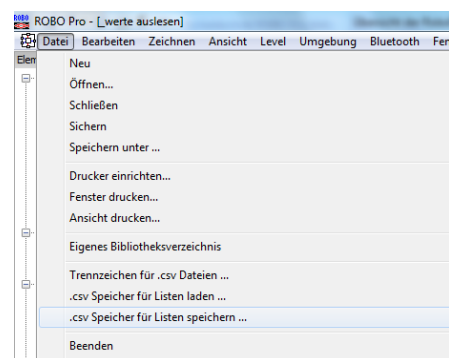


Beispiel 7 Werte über CSV-Datei einlesen



- Dateiname und Pfad angeben
- Spaltennummer angeben
- Spaltenbezeichnung eingeben
- Haken setzen

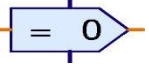
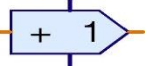
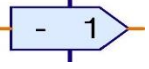





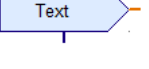
Beispiel 8 Werte in CSV speichern

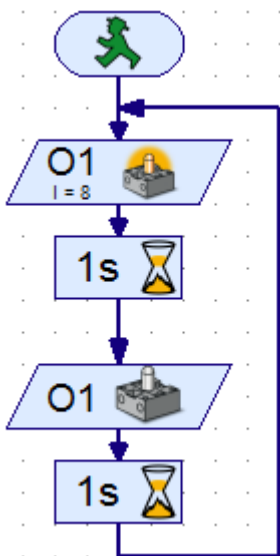


Nach Ende muss explizit gespeichert werden

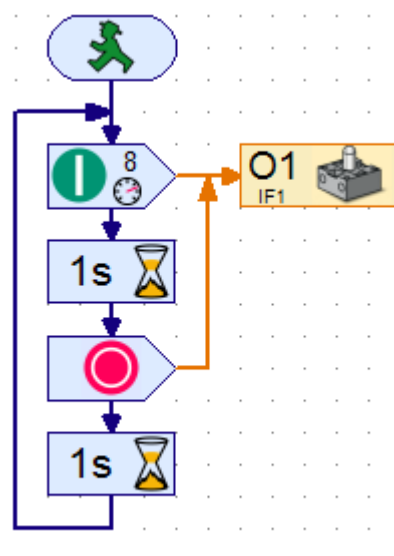
7 Befehle

Befehle werden mit Hilfe von Befehlselementen an die Variablen oder Verbraucher (Motor, Lampe)geschickt. Befehle sind immer an die gelben Datenflußlinien gekoppelt. So erstellte Programme sind Übersichtlicher und Variabler einsetzbar. (Beispiel 1)

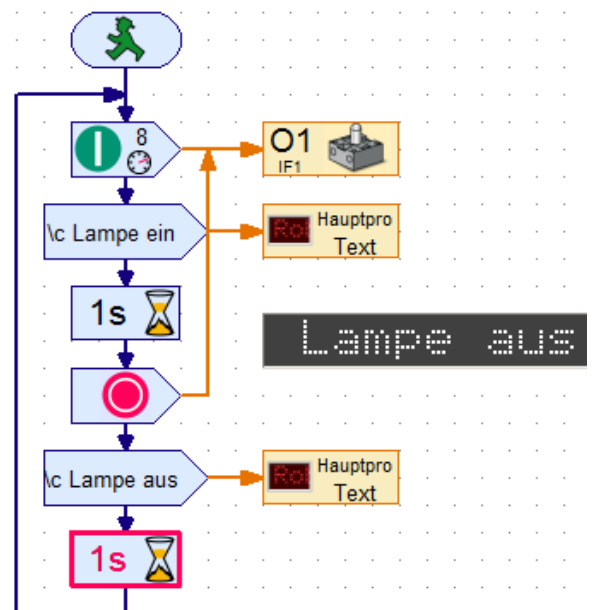
	Befehl Zuweisen	Das Befehlselement Zuweisung sendet den eingestellten Befehl an das angeschlossene Element. (Hier eine Null.) In der Regel wird er verwendet um Variablen, Timer-Variablen, Listenelementen oder Bedienfeld-Ausgängen einen Wert zuzuweisen. (Beispiel 3)
	Befehl Plus	Mit dem +-Befehl kann ein beliebiger Wert übermittelt werden, der zu der Variablen hinzu addiert wird. (Beispiel 4)
	Befehl Minus	Der – Befehl wird ähnlich verwendet wie der zuvor beschriebene + Befehl. Der einzige Unterschied ist, dass der mit dem Befehl übermittelte Wert vom Wert der Variable abgezogen wird
	Befehl Rechts	Der Befehl Rechts wird an Motorausgangselemente geschickt um den Motor mit Drehrichtung rechts einzuschalten. Der Wert ist eine Geschwindigkeit von 1 bis 8. (Beispiel 5)
	Befehl Links	Der Befehl Links wird an Motorausgangselemente geschickt um den Motor mit Drehrichtung links einzuschalten. Der Wert ist eine Geschwindigkeit von 1 bis 8.
	Befehl Stop	Der Befehl Stopp wird an Motorausgangselemente geschickt um den Motor anzuhalten. Mit dem Stopp Befehl wird kein Wert übermittelt.
	Befehl Ein	Der Befehl Ein wird an Lampenausgangselemente geschickt um die Lampe einzuschalten. Der Wert der Helligkeit oder Intensität von 1 bis 8.
	Befehl Aus	Der Befehl Aus wird an Lampenausgangselemente geschickt um die Lampe auszuschalten. Mit dem Aus Befehl wird kein Wert übermittelt.
	Text	Sendet einen Text an ein angeschlossenes Bedienfeld. Der Text wird über das Kontextmenü eingegeben. Vorangestelltes „\c“ löscht den alten Text. (Beispiel 2)



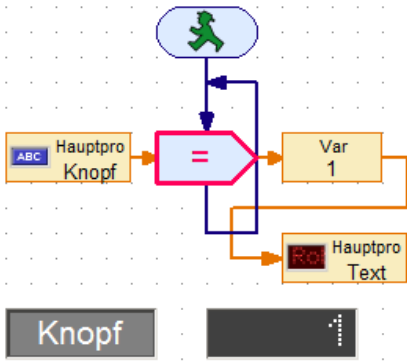
Beispiel 1 (Level 1)



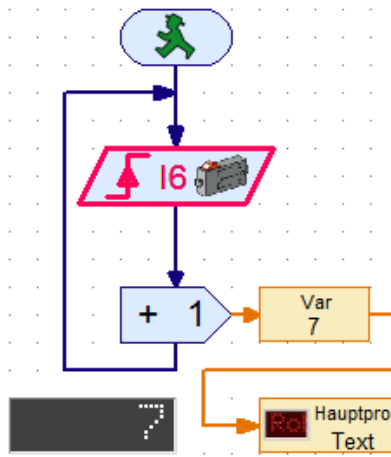
Beispiel 1 (Level 3)



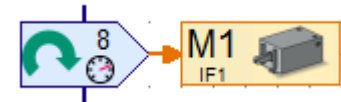
Beispiel 2 (Textausgabe)



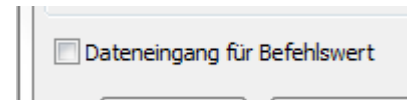
Beispiel 3



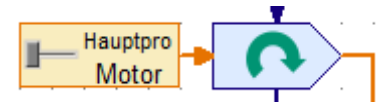
Beispiel 4



Befehl ohne Dateneingang



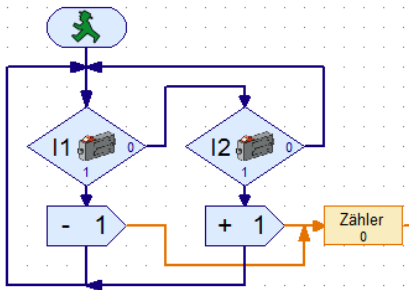
(Aufruf über Kontextmenü)



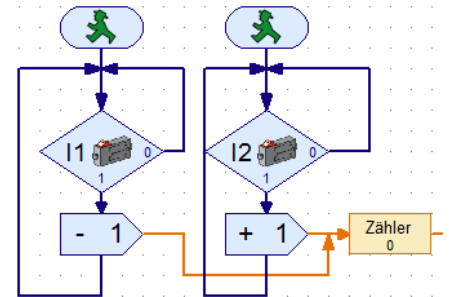
Befehl mit Dateneingang

Beispiel 5

8 Mehrere Prozesse

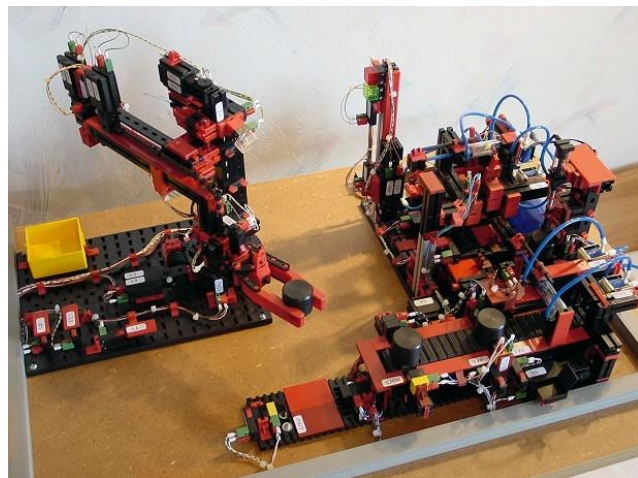
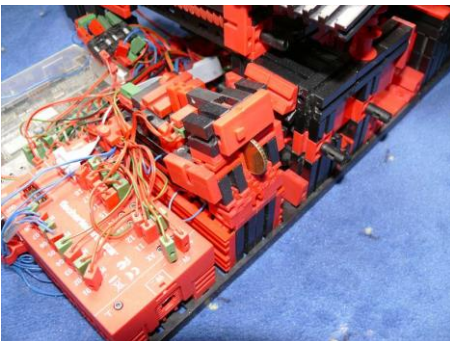


Beim linken Programm können nicht beide Taster gleichzeitig ausgewertet werden. Solange eine Taste gedrückt ist kann das Programm nicht auf den anderen Taster reagieren. Daher ist es möglich, mehrere parallele Prozesse auszuführen. (rechtes Programm)

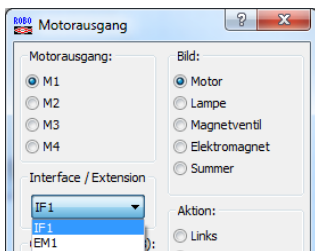
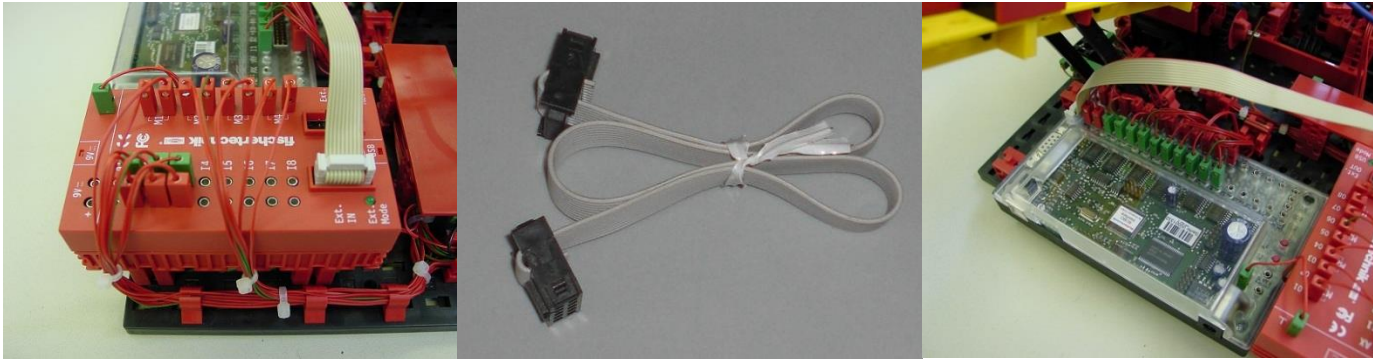


Funktion	Symbol	Bedienfeld	TX Display	Eigenschaften
Name:	Hauptprogramm			
Gruppe:				
Symbolerzeugung:	<input checked="" type="radio"/> Automatisch <input type="radio"/> Manuell			
Standard Platzierung:	<input type="radio"/> Dynamisch <input type="radio"/> Statisch			
Mindestzahl Prozesse	5			
Zusätzliche Prozesse	0			
Mindestspeicher pro Prozess (download)	4096			
Mindestspeicher pro Prozess (online)	65536			

Das wird durch den Einsatz mehrerer Startelemente erreicht (nebenläufige Prozesse). Die Mindestanzahl der Prozesse beträgt 5. Unter dem Reiter Eigenschaften, lässt sich diese Anzahl erhöhen.



9 Erweiterungsmodule



Sollten für eine Steuerung mehr als 8 Eingänge oder 4 Ausgänge benötigt werden, kann das Erweiterungsmodul eingesetzt werden. Es wird über ein 10-Poliges Flachbandkabel mit dem Interface verbunden (*Bild oben*). Bei dem entsprechenden Bauelement wird dabei über das Kontextmenü (*Bild links*) das jeweilige Erweiterungsmodul ausgewählt. Weitere Module können kaskadiert werden. Hinweis: für jedes Modul ist eine eigene Stromversorgung notwendig.

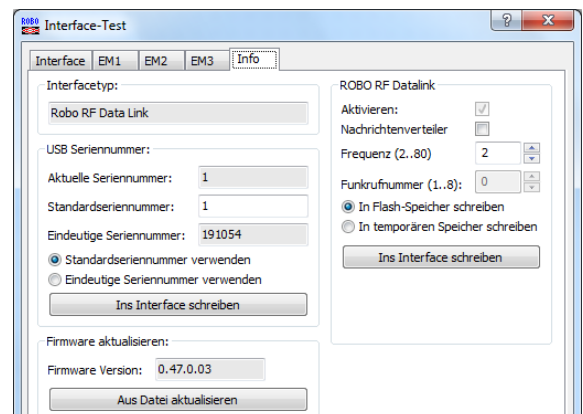
10 Ändern der Interface-Seriennummer

Standardmäßig werden alle ROBO Interfaces und ROBO I/O-Extensions mit der gleichen Seriennummer ausgeliefert.

So ist ein Austausch der Geräte problemlos möglich.

Sollen aber mehrere Interfaces an einem PC über USB betrieben werden, muss eine Seriennummer geändert werden, damit diese unterschieden werden können.

Dazu ist das Fenster „Interface testen“ zu öffnen und der Reiter Info anzuspriegen. Auswahlknopf „Eindeutige Seriennummer verwenden“ anklicken. Anschließend den Knopf „Ins Interface schreiben“ drücken und das Interface kurz vom Strom trennen.



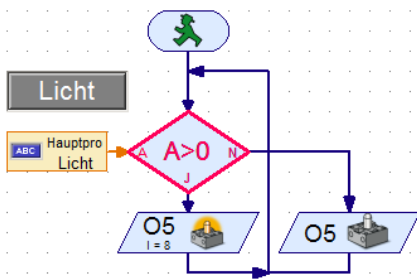
Achtung: Wenn die Seriennummer geändert wurde, muss evt. der Treiber neu installiert werden.

Die Seriennummer kann auch auf 1 zurückgesetzt werden. Dazu ist beim Einschalten des Interfaces die Port-Taste gedrückt zu halten.

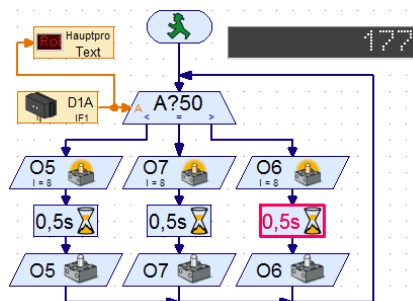
11 Verzweigungen, Warten

Die Programmelemente in dieser Gruppe (Level 3) dienen alle zur Programmverzweigung oder zur Verzögerung des Programmablaufs.

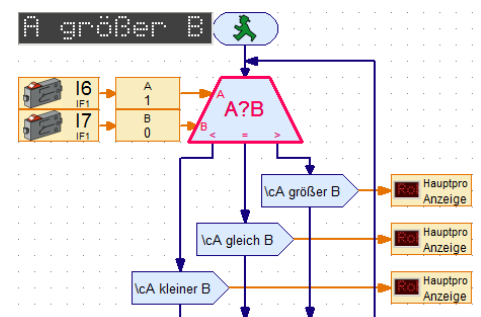
Symbol	Bezeichnung	Erläuterung
	Verzweigung mit Dateneingang	Die Verzweigung vergleicht den Wert vom Dateneingang A mit dem festgelegten Vergleichswert. (Beispiel 1)
	Vergleich mit Festwert	Das Programmelement vergleicht den Wert vom Dateneingang A mit dem hinterlegten Vergleichswert. (Beispiel 2)
	Vergleich	Das Programmelement vergleicht den Wert von Dateneingang A und B. (Beispiel 3)
	Warten auf Zustand 1	Wartet mit dem Programmablauf bis an den Dateneingang der Zustand 1 anliegt. (Beispiel 4)
	Warten auf Zustand 0	Wartet mit dem Programmablauf bis an den Dateneingang der Zustand 1 anliegt. (Beispiel 4)
	Warten auf Änderung 0 oder 1	Wartet mit dem Programmablauf bis an den Dateneingang eine beliebige Änderung erfolgt. (Beispiel 5)
	Warten auf Änderung 0 - 1	Wartet mit dem Programmablauf bis an den Dateneingang eine Änderung von 0 auf 1 erfolgt.
	Warten auf Änderung 1 - 0	Wartet mit dem Programmablauf bis an den Dateneingang eine Änderung von 1 auf 0 erfolgt.
	Warten auf N 0 oder 1 Wechsel	Wartet mit dem Programmablauf bis an den Dateneingang N beliebige Änderungen vorliegen. (Beispiel 6)



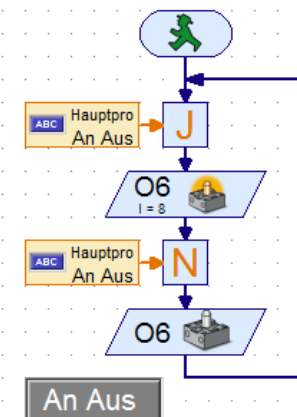
Beispiel 1 (Lampe leuchtet bei gedrücktem Knopf)



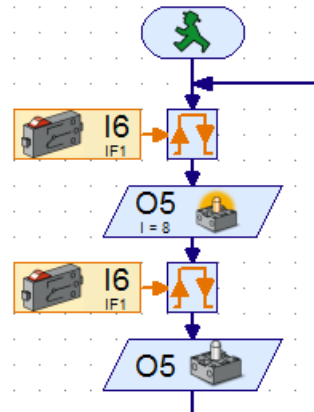
Beispiel 2 (Lampe wird entsprechend dem Abstand eingeschaltet)



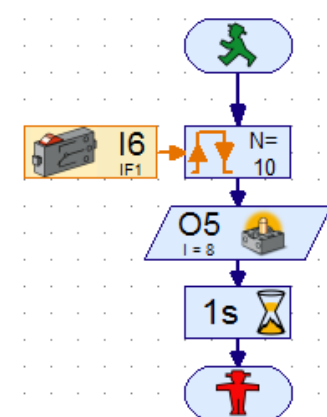
Beispiel 3 (Anzeige entsprechend des Zustandes der beiden Schalter)



Beispiel 4 (Lampe an, solange wie Knopf gedrückt wird)





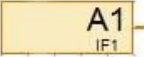
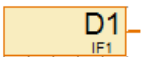
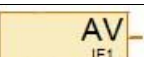

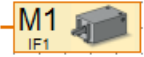
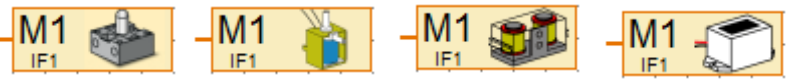




Beispiel 5 (Lampe an, solange wie Schalter gedrückt)



Beispiel 6 (Lampe für eine Sek. an, wenn Schalter 5 mal betätigt)

12 Ein- und Ausgänge

	Digitaleingang: Taster; Fototransistor; Reed-Kontakt, Spuren- Sucher (<i>Beispiel 1</i>)	Anschluss über I1 bis I8; Wartet auf 1 oder 0 	Bei Fototransistor auf Polarität achten! Spurensucher benutzt 2 Digitaleingänge I1 bis I8!
	Analogeingang: NTC-Widerstand; Farb Sensor; Fotowider- stand	Anschluss AX oder AY; Werte werden zwischen 0 und 1023 angegeben 	siehe Tabelle 1
	Spannungsmessung	Anschluss A1 oder A2, Spannungsbereich 0 bis 10 V; Werte werden zwischen 0 und 1023 ausgegeben. siehe Tabelle 2	
	Abstandssensor (Ultraschall)	Anschluss D1 oder D2; Werte werden zwischen 0 und 1023 angegeben; Siehe Tabelle 3 (<i>Beispiel 2</i>)	
	Versorgungsspannung	Gibt einen Wert zwischen 0 und 1023 zurück; Information kommt direkt aus dem Robo Pro, so dass kein Anschluss verfügbar ist. Siehe Tabelle 1	
	Infrarot	Auswertung der Signale der Fernbedienung IP Control Set (<i>Beispiel 3</i>)	
	Motorausgang: Motor, Lampe, Ma- gnetventil; Summer; Elektromagnet	Es werden 2 Anschlüsse verwendet 	
	Lampenausgang: Motor, Lampe, Ma- gnetventil; Summer; Elektromagnet	Es wird nur ein Anschluss verwendet (Zweite gegen Masse) 	

Temperatur	Wert Robo Pro
0 °	880
22 °	400
24 °	370
100 °	100

NTC-Widerstand
Tabelle 1.1

Farbe	Wert Robo Pro
Weiß	160
Gelb	80
Rot	170
Braun	180
Grün	200
Blau	260
Schwarz	300
Grau	100

Farbsensor
Tabelle 1.2
Abhängig von
Entfernung

Helligkeit	Wert Robo Pro
dunkel	1023
Zimmer licht	300
Tages- licht	100
Taschen- lampe	40

Fotowiderstand
Tabelle 1.3

Spannung	Wert Robo Pro
7 V	740
8 V	840
9 V	940
10 V	1023

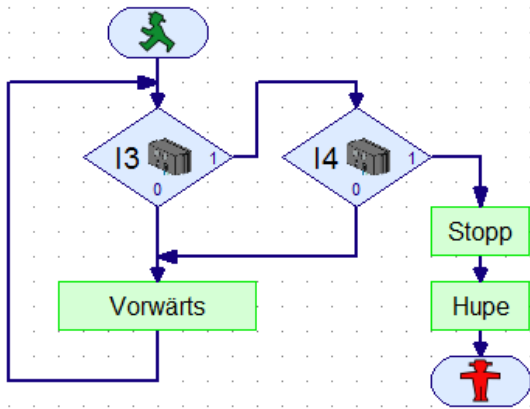
Spannungsmessung
Tabelle 2

Abstand	Wert Robo Pro
10 cm	10
20 cm	20
30 cm	30
40 cm	40
50 cm	50
60 cm	60
70 cm	70
80 cm	80
90 cm	90
100 cm	100

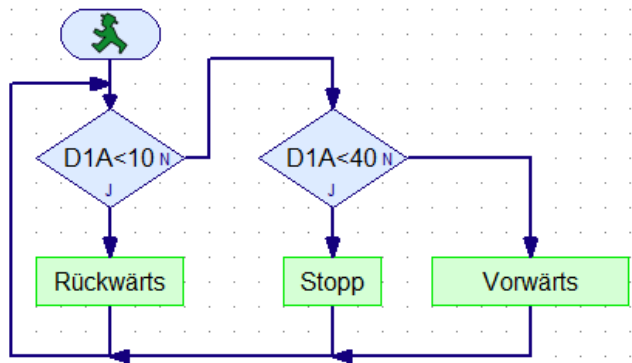
Abstandssensor
Tabelle 3

Anschlüsse:

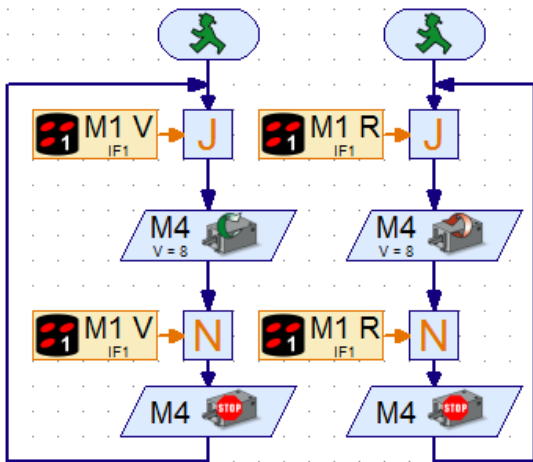
- Farbsensor Rot – Plus; Grün – Minus; Schwarz - A1 oder A2
- Abstandssensor Rot/Grün – D1 oder D2
- Spurensucher Rot – Plus; Grün – Minus; Gelb/Blau – je an I1 bis I8 (Digital)



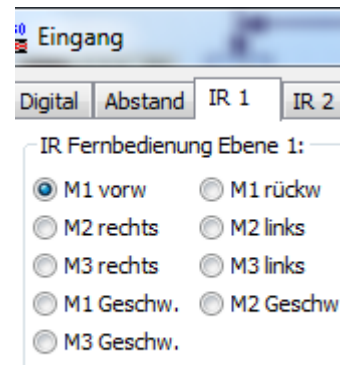
Beispiel 1 (Spurensucher)



Beispiel 2 (Abstandssensor)



Beispiel 3 (Fernbedienung Infrarot)

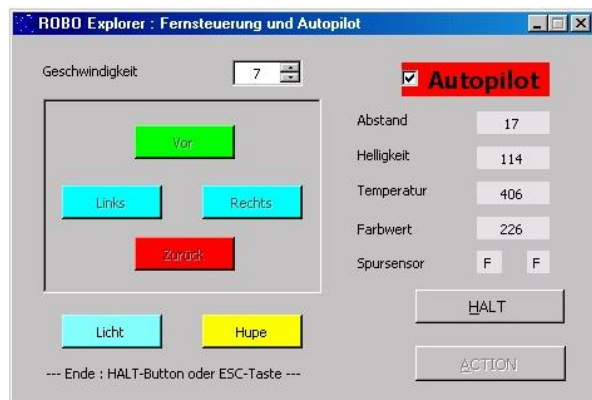
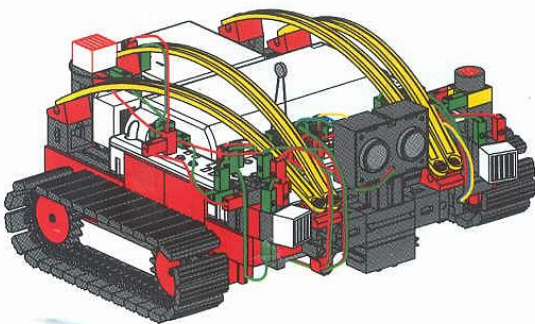


Über die Tasten 1))) und 2))) kann ein weiterer Empfänger gesteuert werden.

Tastenbelegung Sender



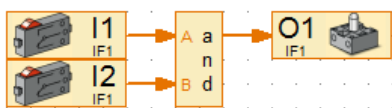
Beispiel zum Auslesen der Werte



13 Operatoren

Logischen Operatoren werten die Eingänge nach 0 oder 1 aus und geben entsprechend 0 oder 1 zurück. Bit Operatoren vergleichen die Eingangswerte Bitweise. Die Rückgabe erfolgt an der jeweiligen Potenz. Arithmetische Operatoren führen ganzzahlige mathematische Berechnungen durch. Vergleichsoperatoren vergleichen die Eingänge auf entsprechende Vorgabe und geben 0 oder 1 zurück.

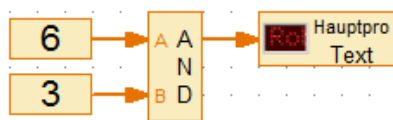
	Logische Operator „Und“ (Beispiel 1)	Der Ausgang ist 1 wenn alle Eingänge auf 1 sind. $0 + 0 = 0 \mid 0 + 1 = 0$ $1 + 0 = 0 \mid 1 + 1 = 1$
	Logische Operator „Oder“	Der Ausgang ist 1 wenn mindestens einer der Eingänge auf 1 ist. $0 + 0 = 0 \mid 0 + 1 = 1$ $1 + 0 = 1 \mid 1 + 1 = 1$
	Logische Operator „Nicht“	Der Ausgang wird invertiert. $1 = 0 \mid 0 = 1$
	Bit Operator „Exclusive Oder“ (Beispiel 2)	Bit Operator „Exclusive Oder“ Ausgang ist 1 wenn nur einer der Eingänge 1 hat. $0 + 0 = 0 \mid 0 + 1 = 1$ $1 + 0 = 1 \mid 1 + 1 = 0$
	Arithmetischer Operator „Plus“ (Beispiel 3)	Bei Division, wird nur der ganzzahlige Teil ausgegeben.
	Vergleichsoperator „Kleiner“ (Beispiel 4)	Gibt 1 zurück, wenn Vergleich wahr, sonst 0.
	Rechnen mit Dezimalzahlen (dazu müssen alle Elemente auf Gleitkomma stehen)	Umwandlung von Dezimalzahlen in Ganzzahlen und umgekehrt <i>Beispiel</i>



Beispiel 1 (logische Operation)

Bei der AND-Operation mit den Zahlen 3 und 6 wurde nur bei der Potenz 2^1 jeweils eine 1 erkannt (grün), so dass im Ergebnis die 2 angezeigt wird.

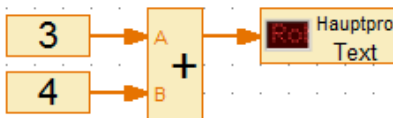
Beispiel 2 (Erläuterung)



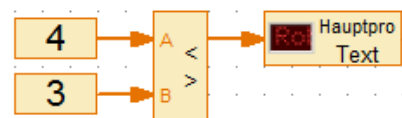
Beispiel 2 (Bit Operatoren)

Potenz	2^3	2^2	2^1	2^0
Wertigkeit	8	4	2	1
6	0	1	1	0
3	0	0	1	1
2	0	0	1	0

Beispiel 2 (Binäre Darstellung)








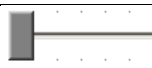


Beispiel 3 (arithmetische Operation)

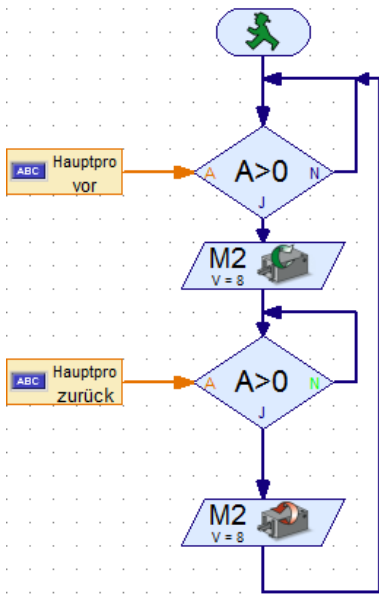


Beispiel 4 (Vergleichsoperation)

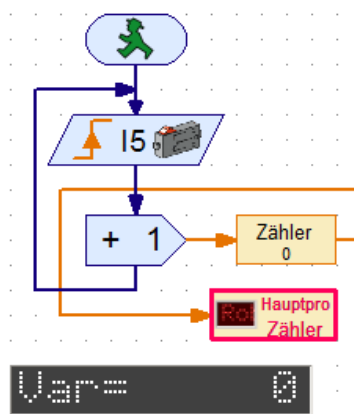
14 Bedienfelder

	Bedienfeldeingang *1	Der Bedienfeldeingang fragt definierte Druckknöpfe und Schieberegler ab. Damit lassen sich (nur!) im Onlinemodus Aktionen über dieses Bedienelement auslösen. (Beispiel 1)
	Bedienfeldanzeige Analog *1	Die Bedienfeldanzeige und der Bedienfeldausgang dienen zur Anzeige von Analogwerten oder Textausgaben. Beides macht nur im Onlinemodus Sinn. (Beispiel 4)
	Bedienfeldanzeige Text *1	Textanzeige (Beispiel 2)
	Analog Anzeige	Anzeige von Analog Werten (Beispiel 4)
	Textanzeige *2	Ausgabe von Zahlen und Texten (Beispiel 2)
	Anzeigelampe *2	Eine Art Lampenbaustein zur Signalisierung eines digitalen Ausgangs (nur Onlinemodus). (Umschaltbar aus Befehlsfeldanzeige) (Beispiel 3)
	Knopf *3	Bedienelement, welches einem Schalter entspricht. (Beispiel 1)
	Regler *3	Bedienelement, welches einem Potentiometer entspricht. (Beispiel 4)

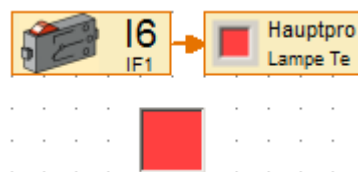
*1 Programmelemente / Ein-, Ausgänge *2 Bedienelemente / Anzeigen *3 Bedienelemente / Steuerelemente



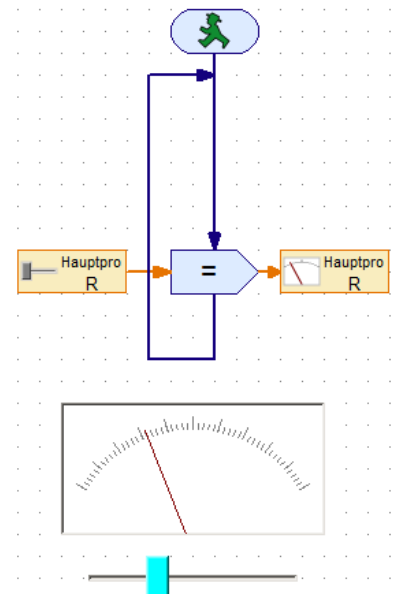
vor
zurück



Beispiel 2



Beispiel 3

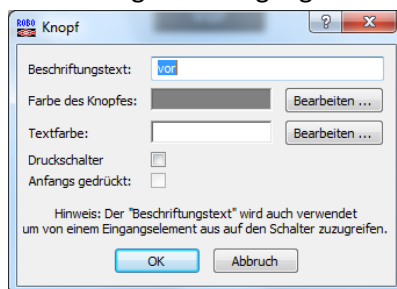


Beispiel 4

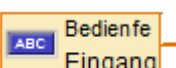
Beispiel 1 Motorsteuerung

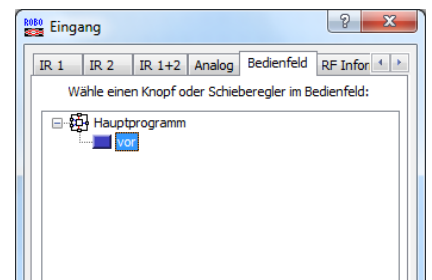
Zusammenhang zwischen Anzeige und Eingang am Beispiel 1


Rechte Maustaste
klicken



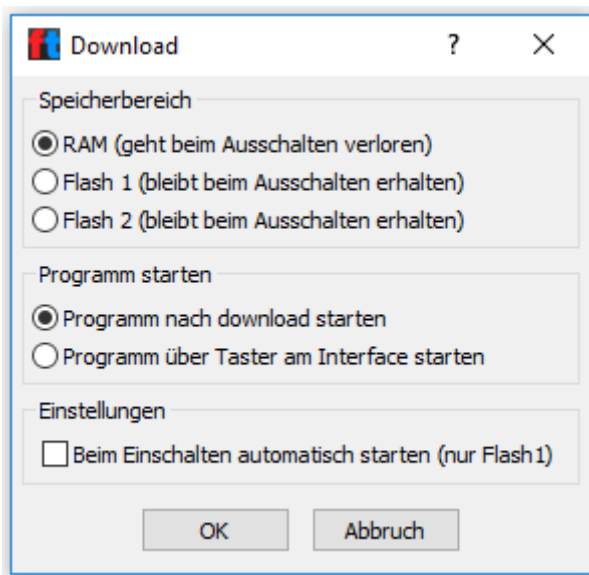
Name für Knopf vergeben



Rechte Maustaste
klicken



Verknüpfung auf Knopf erstellen

16 Programme auf RoboPro laden und starten



Über Taste  wird links stehendes Menü geöffnet. Flash 1 oder Flash 2 auswählen und mit OK bestätigen. Erfolgreiche Übertragung wird angezeigt.

An RoboPro Prog-Taste drücken bis Prog 1 oder Prog 2 angezeigt werden
Prog-Taste erneut kurz drücken bis LED blinkt



15 Robo-Interface LED Error Codes

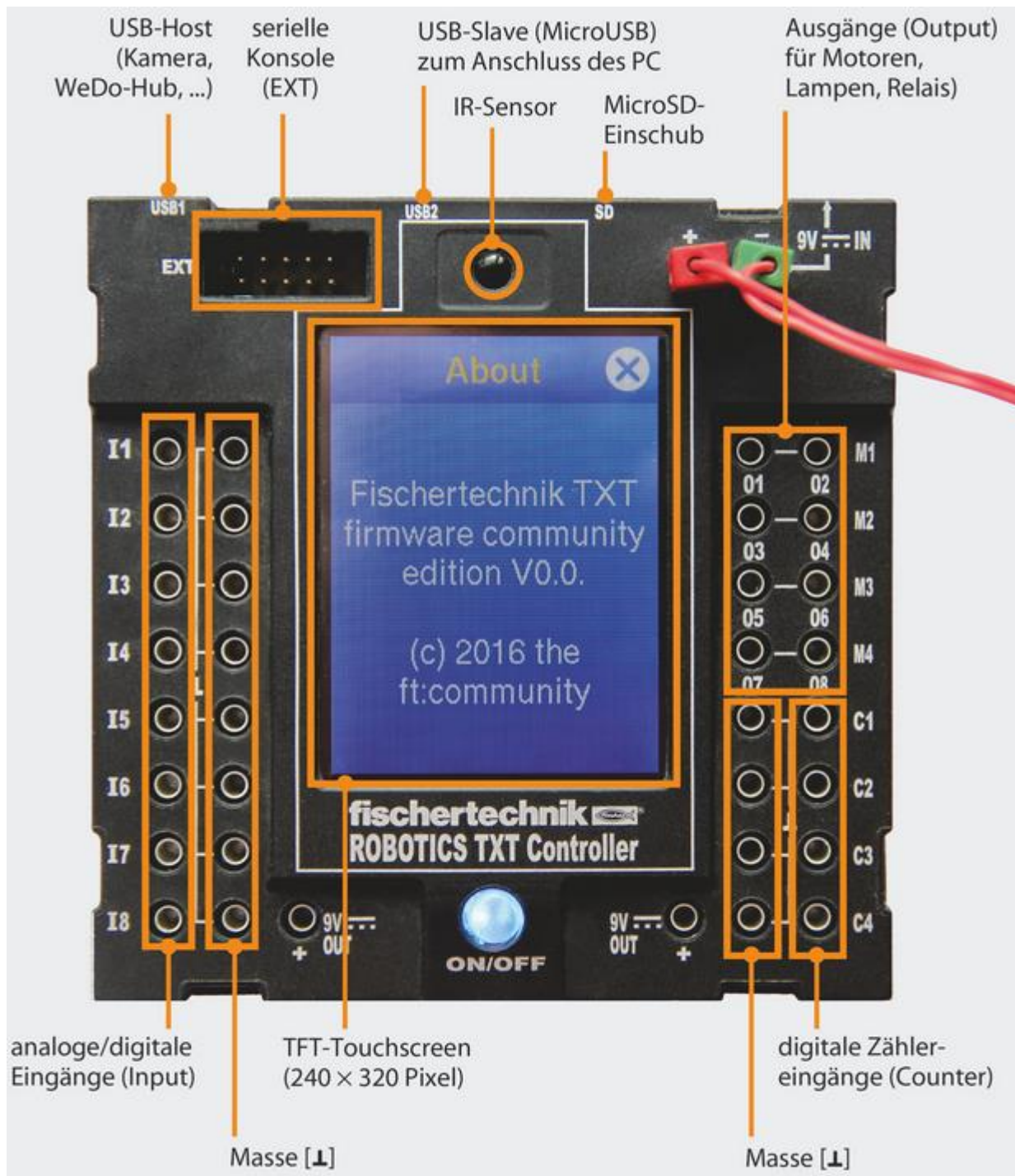
Das Aufleuchten oder Blinken der roten ERROR-LED signalisiert einen schwerwiegenden Fehler. Sie leuchtet dauerhaft, wenn der Prozessor "steht" (z.B. aufgrund zu niedriger Versorgungsspannung). Sie blinkt, wenn der Prozessor diese trotz Fehler noch schalten kann. Wenn die ERROR-LED blinkt, zeigen die LED's für USB, SER und Prog1 / Prog2 einen Fehlercode an:

Error	Prog2	Prog1	USB	SER	IR	Beschreibung
Ein	X	X	X	X	X	Prozessor "steht", Versorgungsspannung evtl. zu niedrig?
Ein	Ein	Ein	Aus	Aus	Ein	Zustand direkt nach dem Einschalten wenn Akku leer ist!
Blinken	aus	aus	aus	aus	ein	Power-On-Error: Interner RAM-Test-1
Blinken	aus	aus	aus	ein	aus	Power-On-Error: Interner RAM-Test-2
Blinken	aus	aus	aus	ein	ein	Power-On-Error: Interner RAM-Test-3
Blinken	aus	aus	ein	aus	ein	Power-On-Error: Ext. RAM-Test-1
Blinken	aus	aus	ein	ein	aus	Power-On-Error: Ext. RAM-Test-2
Blinken	aus	aus	ein	ein	ein	Power-On-Error: Ext. RAM-Test-3
Blinken	aus	ein	aus	aus	aus	Power-On-Error: Bootloader: Kontakt zum Service aufnehmen!
Blinken	aus	ein	aus	aus	ein	Power-On-Error: Firmware: Kontakt zum Service aufnehmen!
Blinken	aus	ein	aus	ein	aus	Power-On-Error: Versorgungsspannung für "NOTBETRIEB-Firmwareupdate" ist zu niedrig
Blinken	aus	Blinken	aus	aus	ein	Power-On-Error: Firmware: Durch gleichzeitigen Tastendruck von "Port" und "Prog" kann letzter Firmwareupdate wiederholt werden ("NOTBETRIEB Firmwareupdate"). Zum Starten ca. 2 Sekunden die Taster drücken bis IR-Led blinkt, dann loslassen
Blinken	aus	Blinken	aus	aus	Blinken	Power-On-Error: Firmware: Durch gleichzeitigen Tastendruck von "Port" und "Prog" kann Flashen wiederholt werden, nun Taster loslassen zum Starten des Firmwareupdate im Notbetrieb
Blinken	aus	aus	aus	aus	Blinken	Programmfehler: Unbekannter Befehl-1
Blinken	aus	aus	aus	Blinken	aus	Programmfehler: Unbekannter Befehl-2
Blinken	aus	aus	aus	Blinken	Blinken	Programmfehler: Unbekannter Befehl-3
Blinken	aus	aus	aus	aus	aus	Firmwareupdate: Fehler beim Flash löschen
Blinken	aus	aus	Blinken	aus	aus	Interner Fehler I1
Blinken	aus	aus	Blinken	aus	Blinken	Interner Fehler I2
Blinken	Blinken	Blinken	aus	aus	X	Firmware-Update Fehler Kontakt zum Service aufnehmen!

X = Zustand unbestimmt

I TXT-Controller

I.I Anschlüsse



Im Jahr 2013 kam der Fischertechnik TXT Controller auf den Markt

Prozessor: TI AM3359 32-bit, ARM Cortex-A8, 600 Mhz;

Speicher: 256 MB RAM / 128 MB Flash

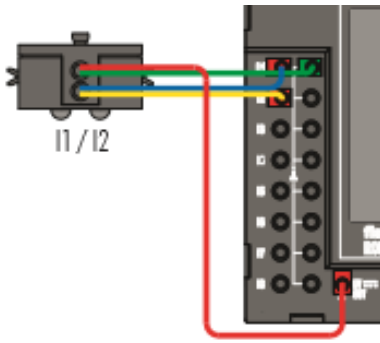
Display: 240x320 Pixel, farbig, 2,4", Touchscreen;

Größe: 90 mm x 90 mm x 25 mm

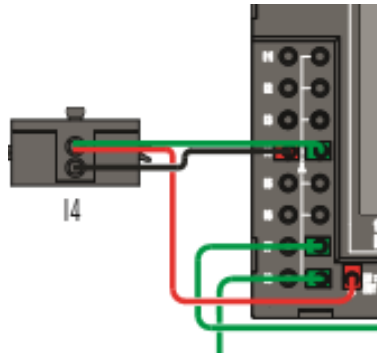
Ein-/Ausgänge: 8 Universaleingänge, 4 Zähleringänge, 4 Motorausgänge, alternativ 8 Einzelausgänge

Schnittstellen: Bluetooth 2.1, I²C, Infrarotschnittstelle, WLAN 802.11, Lautsprecher, microSD- Slot

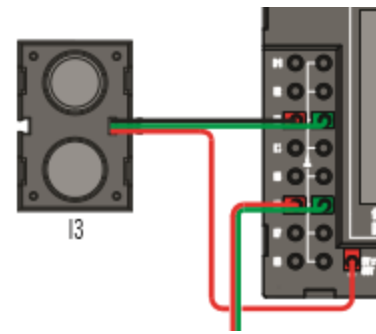
I.II Anschluß von Sensoren



Spurensucher
(Digital 10 Volt)

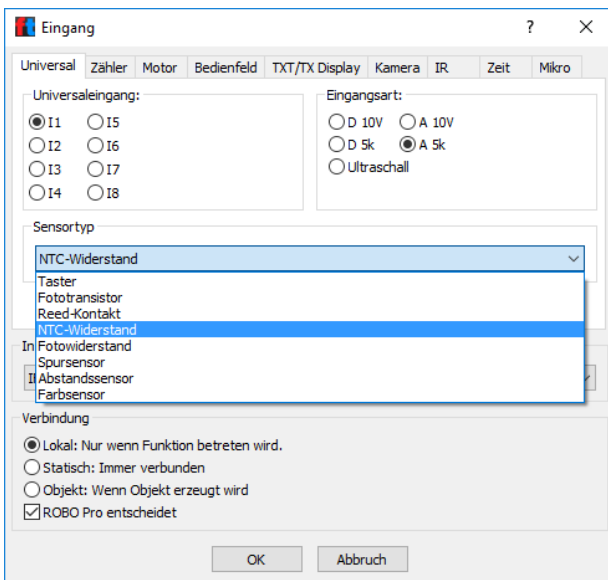


Farberkennung
(Analog 10 Volt)

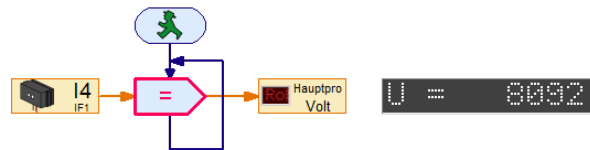


Abstandssensor
(Ultraschall)

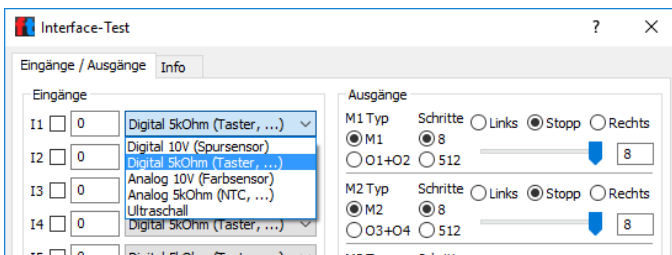
I.III Universaleingänge umschalten



Entsprechend den Erfordernissen, können die Eingänge umgeschaltet werden. Standard ist „Digital 5KOhm“, um einen Schalter abzufragen. Für Spannungsmessungen (siehe Beispiel) wird „Analog 10V“ benutzt.



Für NTC = 5KOhm
Für Fotowiderstand = Analog
Für Fototransistor = Digital
Für Entfernung = Ultraschall



Auch beim Interface-Test kann der Universaleingang umgeschaltet werden.

I.IV Weitere Eingangsmöglichkeiten

Zähler – Digital oder Analog

Motor – Motor Steuereingang (Position erreicht)

Bedienfeld – Vorher ein Bedienelement / Steuerelement anlegen und auswählen

TXT Display – Vorher ein TX Controller Displayelement anlegen und auswählen

Kamera – Vorher Bedienelement / Kamerabildschirm anlegen und auswählen

IR – für Fischertechnik Control Set (siehe 3.3)

Zeit – Auswertung von Sekunde, Minute, Stunde, Wochentag, Tag, Monat, Jahr

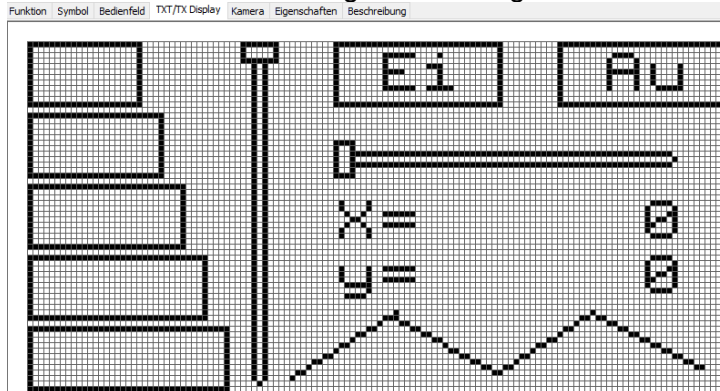
Mikro – Eingang verwendet das Mikrophon der USB-Kamera

II.Display

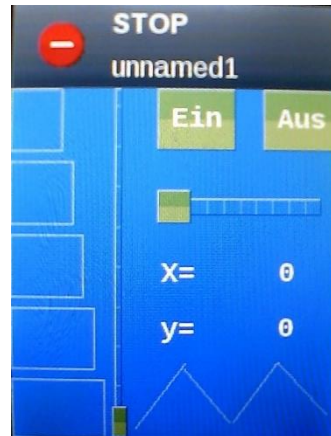
Das farbige Touch Display zeigt den Status des Controllers, welche Programme geladen sind und wo man sich im Menü befindet. Es lassen sich Funktionen und Programme auswählen, aktivieren oder deaktivieren. Während ein Programm läuft kann man sich Werte von Variablen oder Werte von analogen Sensoren anzeigen lassen.

II.I Größenanpassung

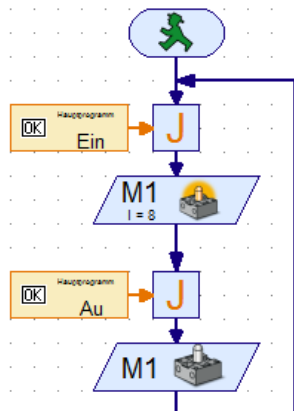
- Die Größen der Eingabe entsprechen nicht der Darstellung auf dem Display
- Steuerelemente können nicht in der Größe verändert werden
- Die Farbdarstellung kann nicht geändert werden.



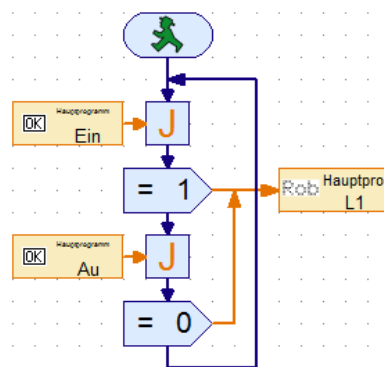
Rechtecke 12*21; 12*25; 12*29; 12*33; 12*37



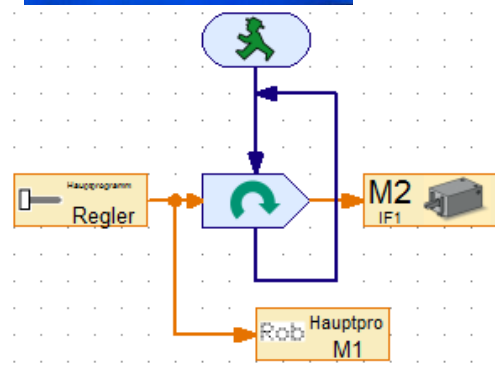
II.II. Steuerung und Anzeige über Display



1. Ein / Aus

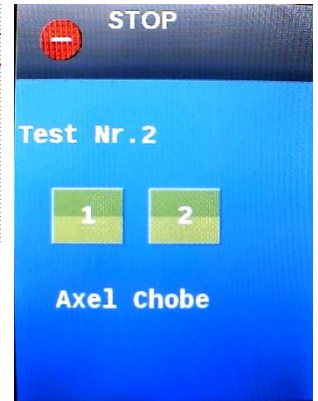
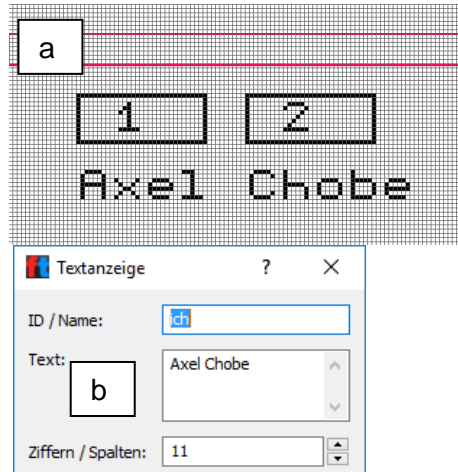
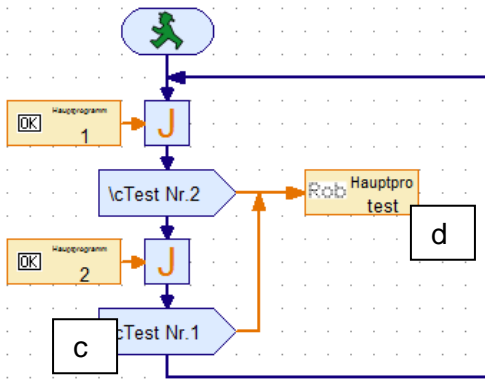


2. Zustand



3. Geschwindigkeit

II.III Textausgabe



Variante 2

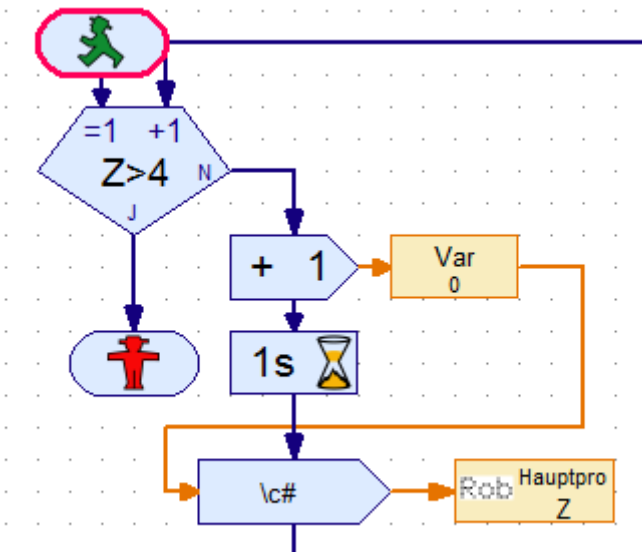
Variante 1

Darstellung 1 und 2

Variante 1: ein Anzeigeelement (a) im Reiter TXT-Display anlegen und unter Eigenschaften (b) mit einem Text versehen (statisch)

Variante 2: den Textbefehl (c) aufrufen und der Bedienfeldanzeige (d) zuweisen (dynamisch)

Variante 3: Ausgabe von Variablen mit dem Textfeld



(Im Textfeld muss der Dateneingang aktiviert werden um die Variable mit „#“ dazustellen)

Variante 3

III.Encodermotor

Encoder-Motoren haben Zusätzlich zum Anschluß für die Stromversorgung eine Buchse für ein 3-poliges Anschlußkabel, über das man die Drehbewegung des Motors auswerten kann. Ein Magnet kommt bei jeder Umdrehung an einen Sensor vorbei, wodurch ein Impuls erzeugt wird. Der Motor hat ein Getriebe mit einer Übersetzung von 21:1, so das 63 Impulse einer Umdrehung der Welle entspricht.

III.I Steuerung

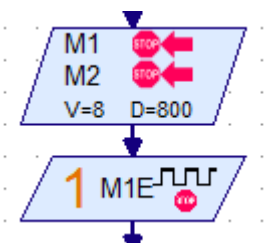


Aktion:
 Abstand – ein Motor eine bestimmte Distanz
 Synchron – zwei Motoren mit gleicher Geschwindigkeit
 Synchron Distanz – zwei Motoren mit bestimmter Distanz
 Stopp – eine Aktion abbrechen


Motorausgang 2:
 Wird als Aktion Synchron gewählt, kann hier der zweite dazugehörige Motor ausgewählt werden.

Geschwindigkeit:
 Kann zwischen 1 und 8 gewählt werden; gilt für beide Motoren

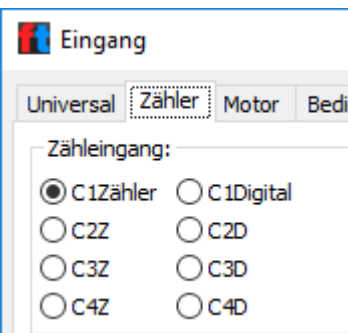
Distanz:
 Anzahl der Encoderimpulse



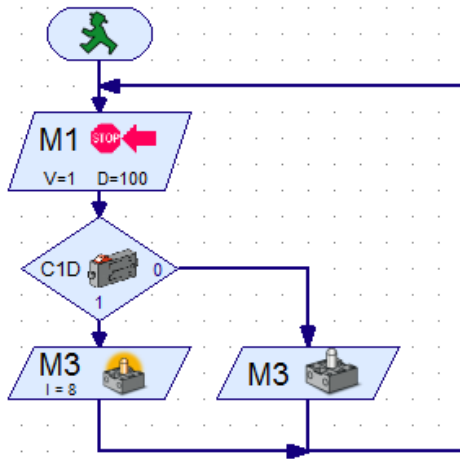
Darstellung Encodermotor (Level1)
 2 Motoren; Drehung links; Geschwindigkeit 8; 800 Impulse
 Hinweis 63 Impulse ergeben eine Umdrehung!

Darstellung „Warten auf Eingang“  Umschalten auf ME 1
 Das Programm wartet nicht auf die Impulse des Motors – hier wird auf die Ausführung gewartet und dann das Programm weitergeführt.

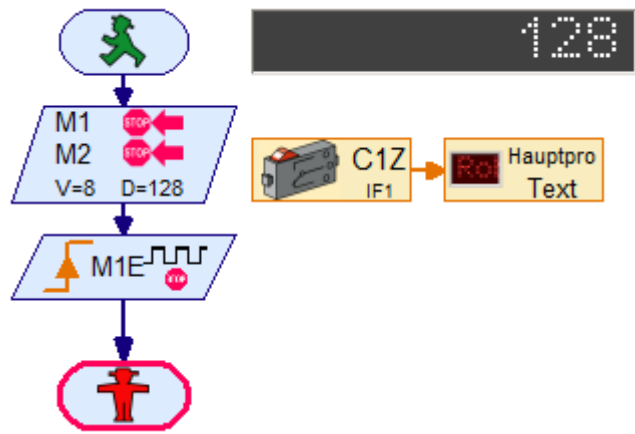
III.II Zähleingänge (Analog und Digital)



Dem Motor M1 wird intern automatisch der Zähleingang C1 zugeordnet, zu M2 gehört C2 .
 Es gibt 4 Zähleingänge für analoge und digitale Auswertung.

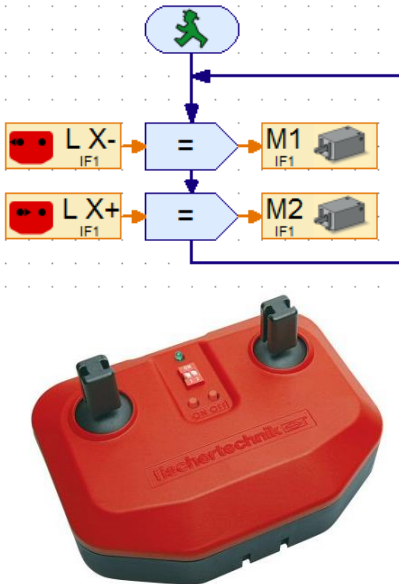


Auswertung digitaler Zählleitung

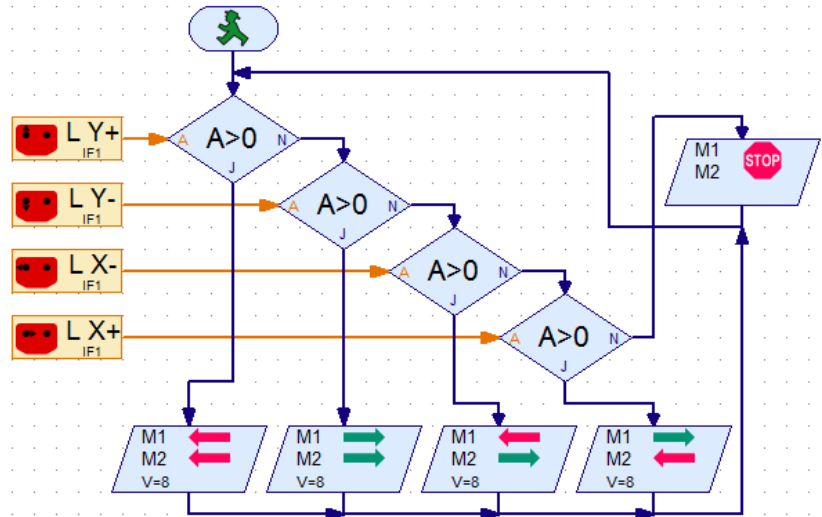


Auswertung analoger Zählleitung

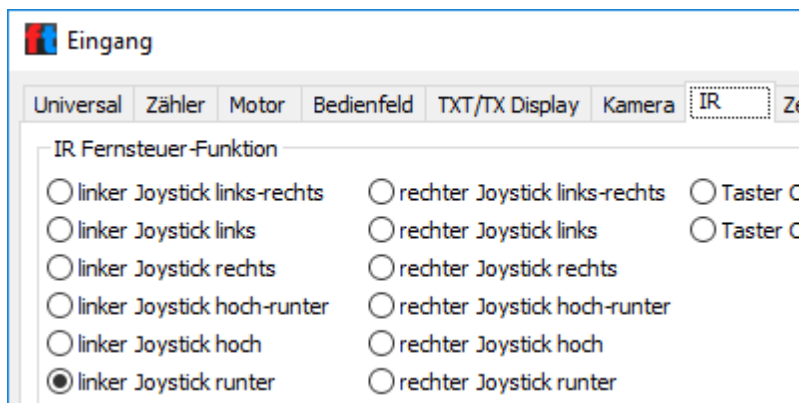
III.III Steuerung mit Fernbedienung Control Set



Einfache Steuerung

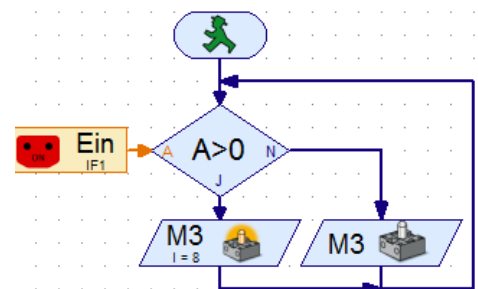


Komplexe Steuerung



Nach Auswahl eines Eingangselementes kann über die Eigenschaften der Reiter IR aufgerufen werden, um die Schalterstellung auszuwählen.

Die Tasten On und Off auf der Fernbedienung können ebenfalls in die Programmierung einbezogen werden.



III.IV WLAN Client-Modus einschalten



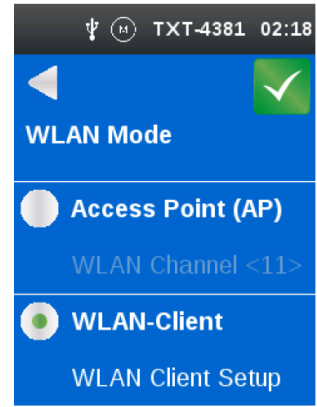
Einstellungen



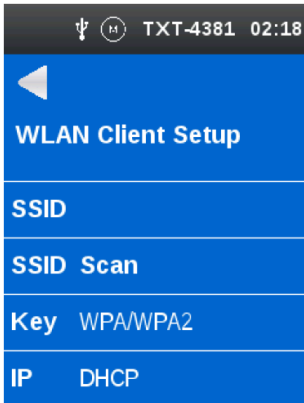
Netzwerk



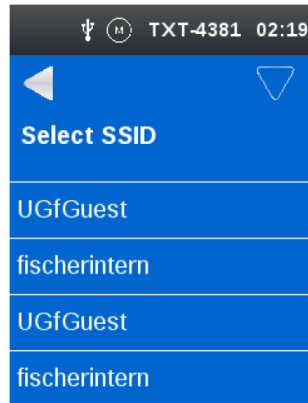
WLAN Mode



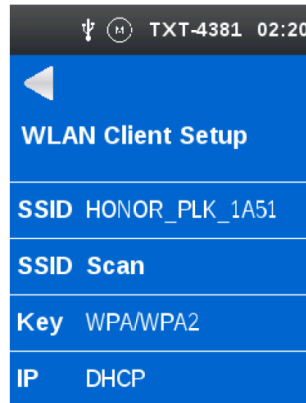
WLAN-Client
WLAN-Client Setup



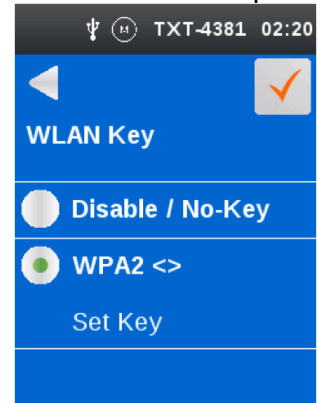
SSID Scan



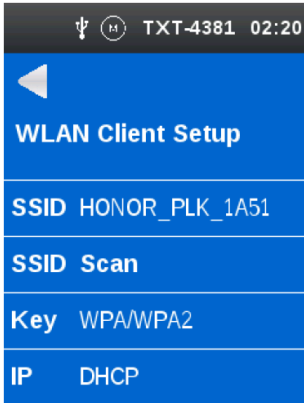
Eigenes Netzwerk
wählen



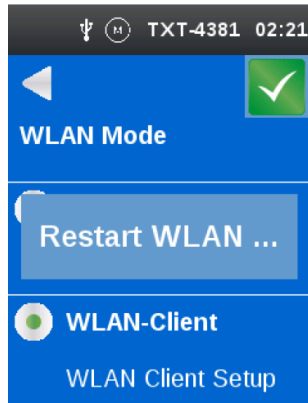
Key



WPA2 Set Key
Schlüssel eintragen



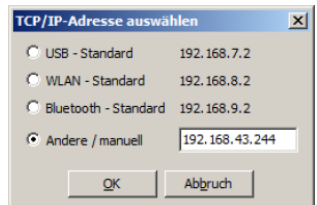
Pfeil nach rechts
drücken



Grünen Haken drücken

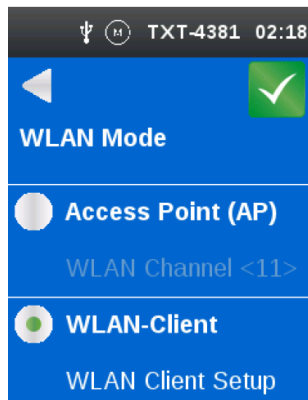


WLAN-Symbol und IP
zeigen das alles OK ist

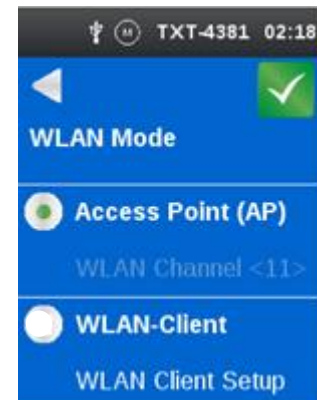


IP-Adresse in RoboPro
zur Drahtlosen
Verbindung eintragen

Diese Einstellung wird benötigt, um mit der RoboPro Software Verbindung aufzunehmen

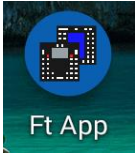


Diese Einstellung wird benötigt, um mit den Fernbedienungs-Programmen auf dem Handy Verbindung aufzunehmen

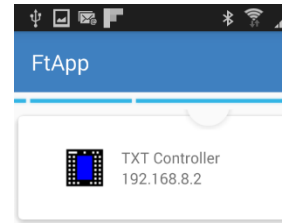


Fertig

III.V Steuerung mit Fernbedienung Handy



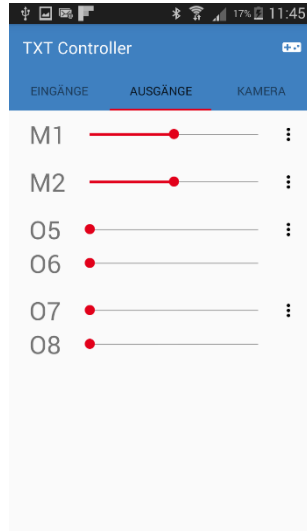
- Die App „Fischertechnik FT App“ auf das Handy laden.
- Handy per Bluetooth mit dem TXT Controller verbinden.
- Bluetooth einschalten
- App starten und Controller wählen



Variante 1



Anzeige der Eingänge



Steuerung der Ausgänge wahlweise Motor od- Lampe

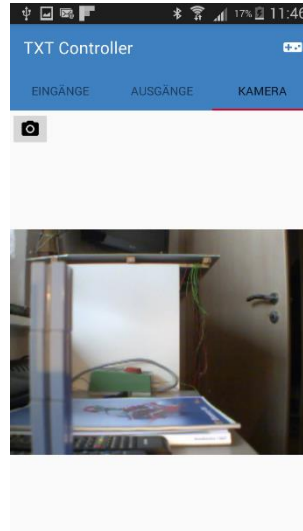
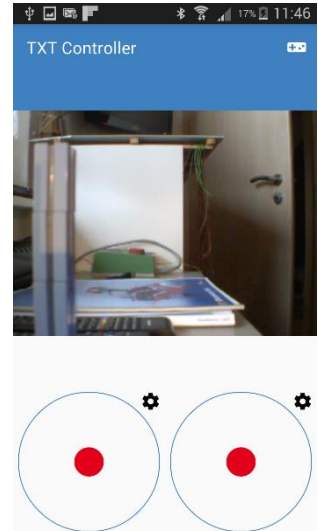


Bild der Kamera

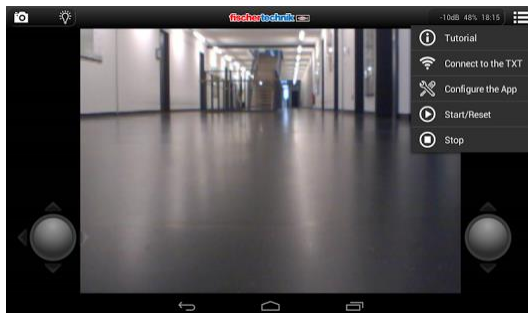


Steuermodus links Motor 1 und 2 rechts Motor 3 und 4

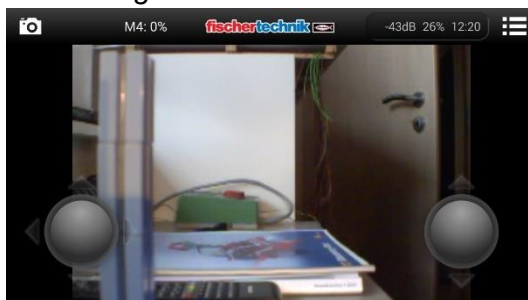


- Die App „TXT Camdroid“ auf das Handy laden
- Das Handy per WLAN mit dem TXT Controller verbinden
- Bluetooth einschalten
- App starten

Variante 2



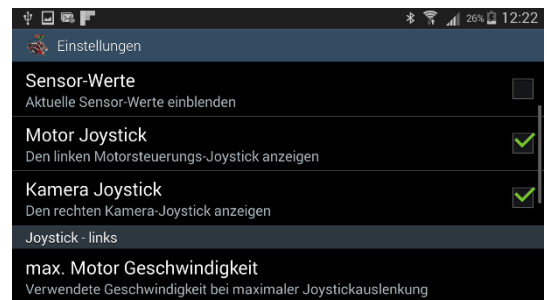
Einstellungen/ Verbinden



Links Motor 1 und 2; rechts Motor 3



WLAN wählen und verbinden



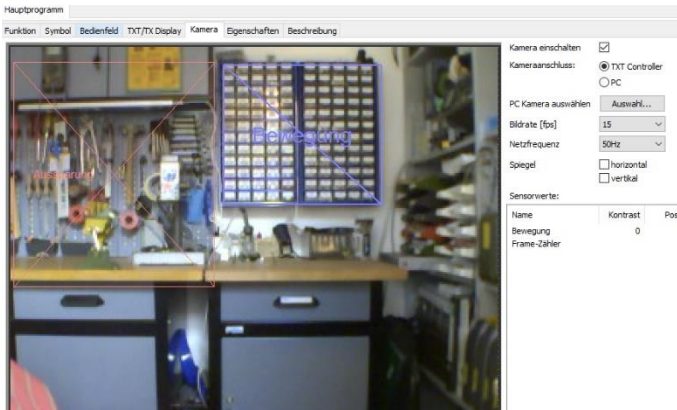
Es sind weitere Einstellungen möglich

IV Kamera

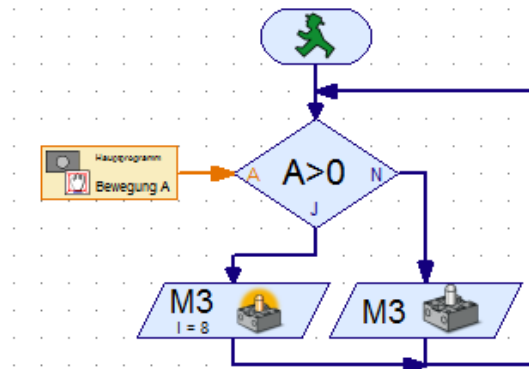
Die Bildauflösung beträgt 1 Megapixel. Die Bilder können an den PC übertragen werden. Der Controller kann die Bilder verarbeiten und dadurch Bewegungen, Farben oder Spuren erkennen.

IV.I Bewegung

Unter dem Reiter Kamera die Bewegungsfläche einstellen

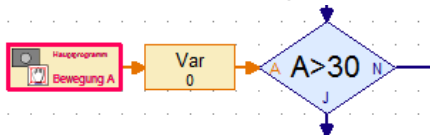


Über den Schubkästen liegt das Feld Bewegung



Wird eine Bewegung im Bereich erkannt, Leuchtet M3

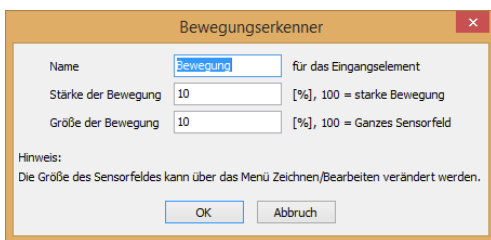
Zum Einstellen der Empfindlichkeit kann die Bewegung über eine Variable abgefangen werden.



(Programelemente/ Verzweigungen)

Das Element stellt 2 Eingangsfelder zur Verfügung, C und A.

- C ist der mittlere Änderungskontrast. Für die Ermittlung des Werts werden nur Bildpunkte berücksichtigt werden, die über dem eingestellten Schwellwert liegen.
- A gibt die Änderungsfläche an, also den Anteil der Fläche in %, in dem der Schwellwert für den Änderungskontrast C überschritten ist.



- Wenn der eingestellte Wert für die Änderungsfläche A nicht erreicht wird, ist der Wert für den Änderungskontrast 0. Wenn man also nur erkennen möchte, ob eine Bewegung erkannt wird, kann man einfach den C Eingang mit 0 vergleichen. Die Werte werden im Eigenschaftsfenster verändert (Reiter Kamera/ rechte Maustaste auf Bereich)

IV.II Farberkennung



Das Element stellt 4 Eingangswerte zur Verfügung, R, G, B und S/W, für die Anteile an den Grundfarben rot, grün und blau und die Gesamthelligkeit in % (Wertebereich 0 bis 100, 0=dunkel bzw. schwarz, 100=hell bzw. weiß).



Der Einsatz der Kamera zur Farberkennung ist nur bedingt geeignet.

IV.III Linienerkennung

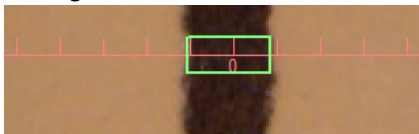
Dieses Element erkennt Linien, die die Linie des Sensorelements kreuzen. Das Element erkennt die Position, Breite und Farbe der Linie.



Linie links vom Mittelwert
Übergabewert -40



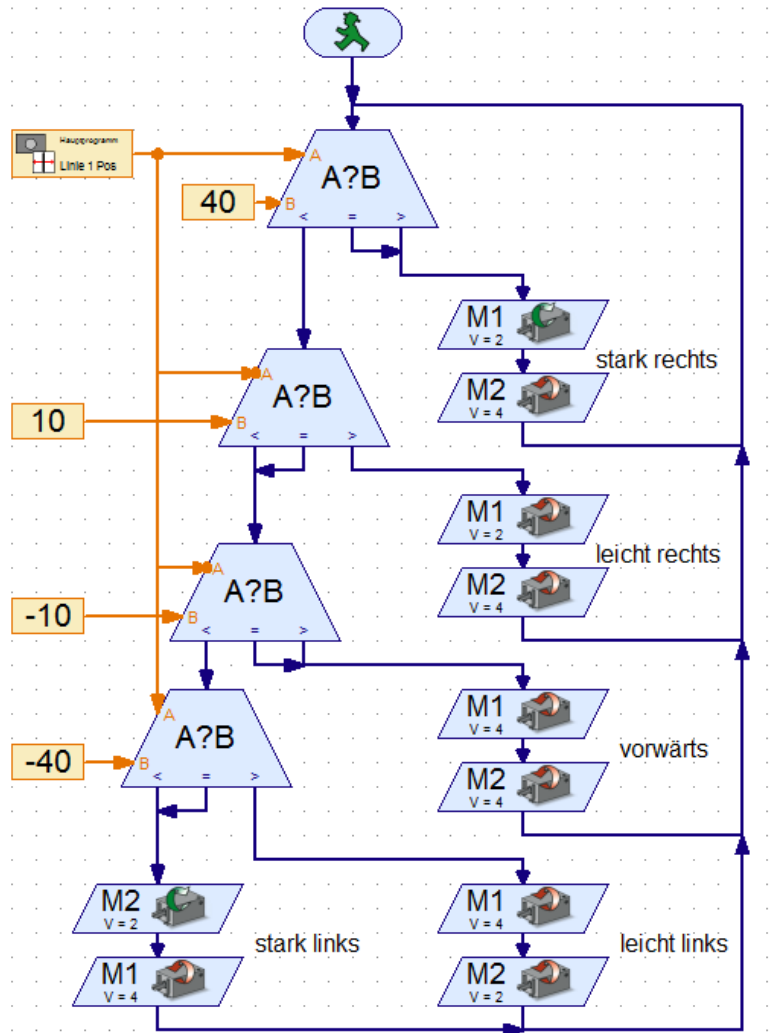
Linie rechts vom Mittelweg
Übergabewert +40



Linie auf Nullpunkt Übergabewert
gegen 0

Mögliche Einstellungen:

- min/max Linienbreite
- min/max Wert (Koordinaten)
- minimaler Kontrast
- Schrittweite



Einfache Motorsteuerung

IV.IV Ballerkennung

Dieses Element erkennt farbige Kreisflächen, Bälle oder andere kompakte farbige Gegenstände vor einem weiß/grau/schwarzen Hintergrund und liefert die Größe und Position des Objekts. Damit dieses Element funktioniert, darf es nur ein farbiges Element im Erkennungsbereich geben.

Mögliche Einstellungen:

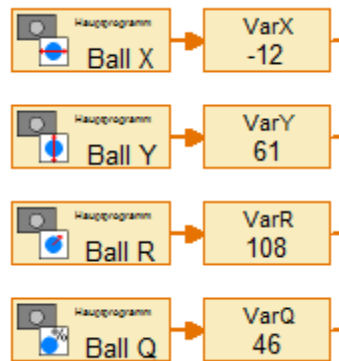
- minFarbkontrast
- min/max Ballgröße
- Aussparungen
- min/max X-Koordinate
- min/max y-Koordinate

Übergabewerte Sensor:

- Ballposition X
- Ballposition Y
- Ballgröße
- Kontrast



Position Ball

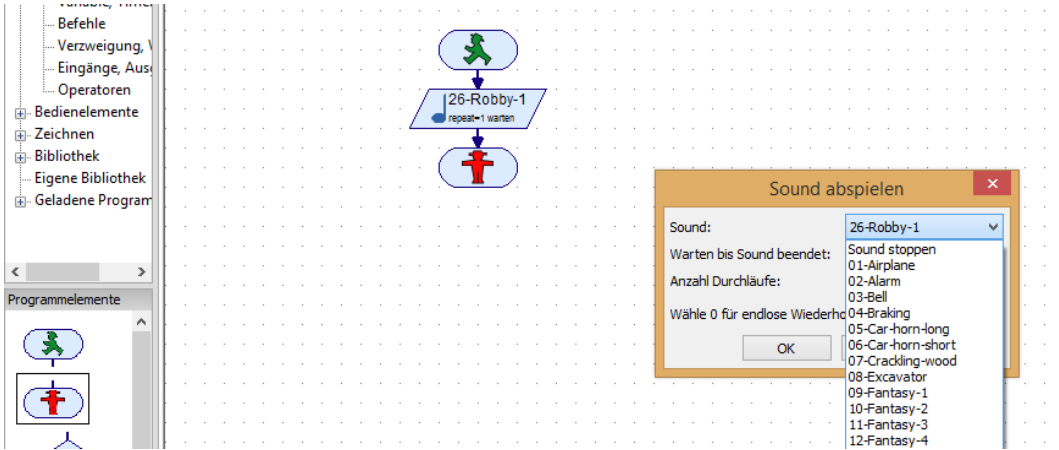


Werte der Ballposition

IV.V. Aussparung

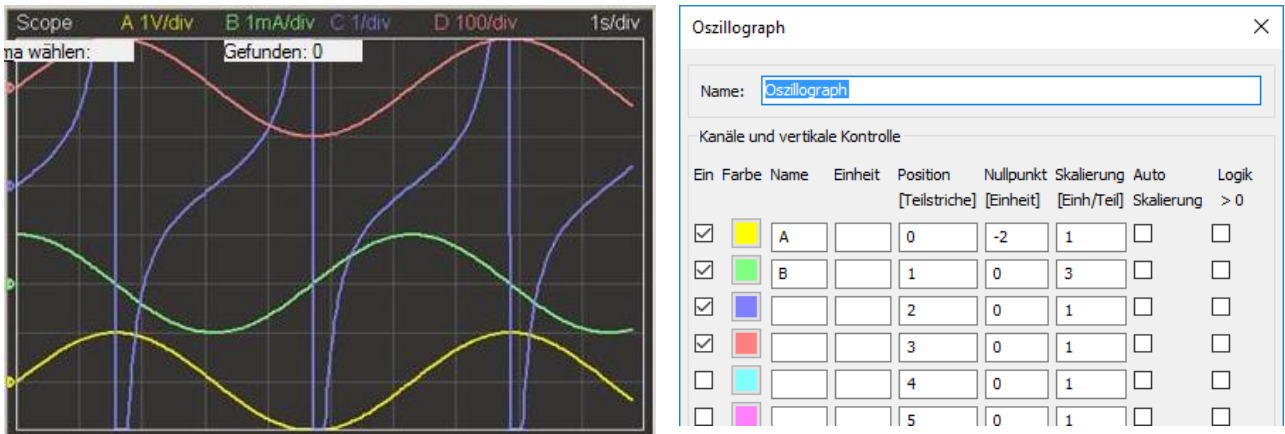
Dieses Element dient dazu, Bereiche aus dem Erkennungsbereich eines Ballerkenners auszublenden, wenn dort z. B. farbige Teile des Modells zu sehen sind, die irrtümlich als Ball erkannt werden könnten.

V Soundausgabe

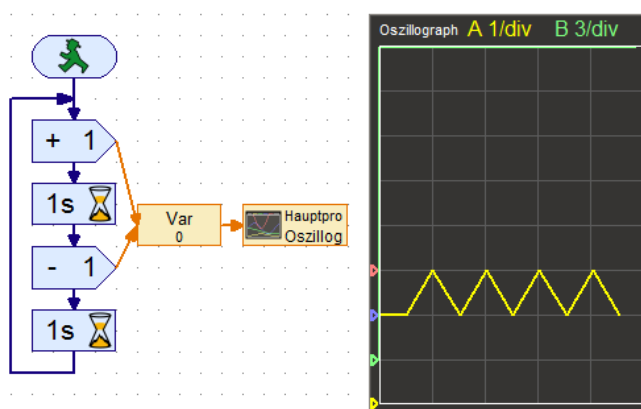


Symbol aus Programmelemente / Grundelemente/ Entsprechenden Sound auswählen

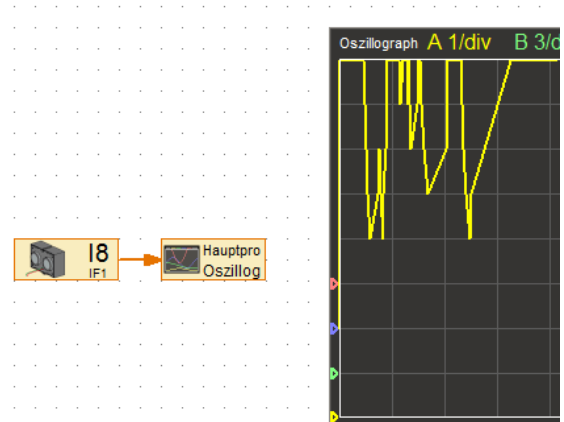
VI Oszillograph (Scope)



Eine Oszillograph, oder kurz Scope, kann den zeitlichen Verlauf von bis zu 8 Eingängen, Variablen oder anderen Datenwerten anzeigen. Es wird über Elementgruppe Eingänge/Ausgänge als Bedienfeldausgang aufgerufen.



Beispiel digitale Auswertung



Beispiel analoge Auswertung

VII Scratch

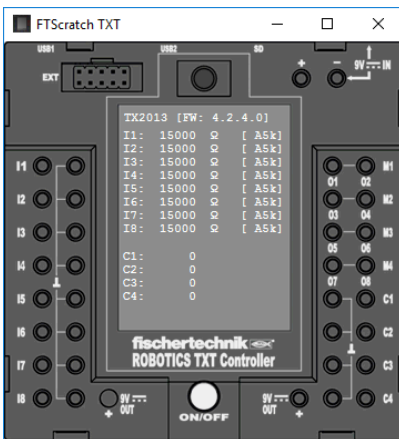


FTScratchTXT ermöglicht auf PCs bzw. Notebooks die Verwendung des Fischertechnik ROBOTICS TXT Controller zusammen mit der einfach zu erlernenden Programmiersprache Scratch. So können im Hand-umdrehen kleine Anwendungen zur Steuerung von Motoren und Lampen mittels Tastern, Lichtschranken, und vielen weiteren Sensoren erstellt werden. Durch die Vielzahl an Möglichkeiten, die die Programmiersprache Scratch bietet, können auch weitaus größere Programme erstellt werden.

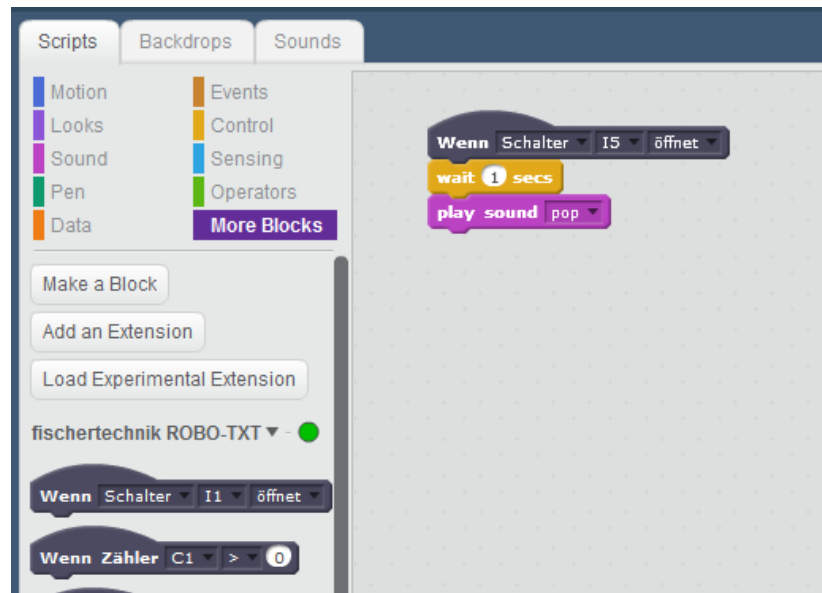
- Controller mit PC verbinden
- FTScratchTXT herunterladen und starten 1*
- ScratchX im Browser öffnen 2*

1* <https://ftscratch.github.io/ROBO-TXT/bin/FTScratchTXT.exe>

2* <http://scratchx.org/?url=http://ftscratch.github.io/ROBO-TXT/src/ext.js#scratch>

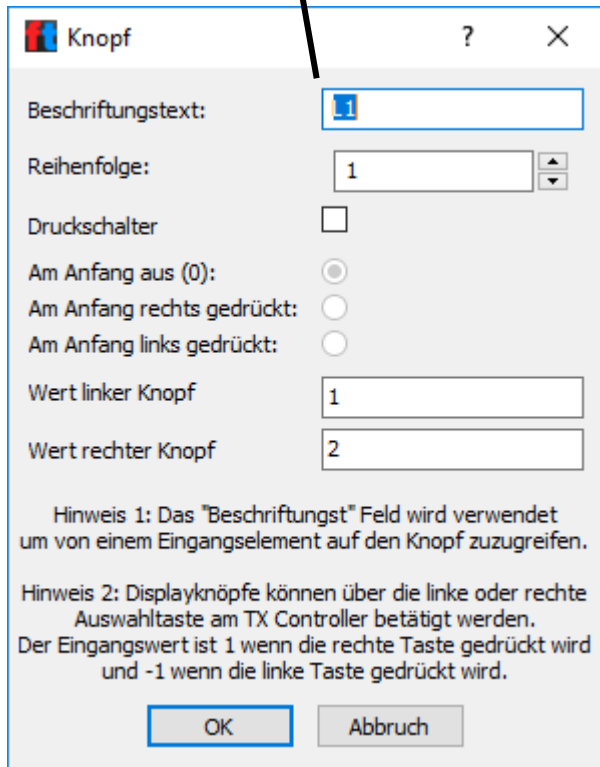
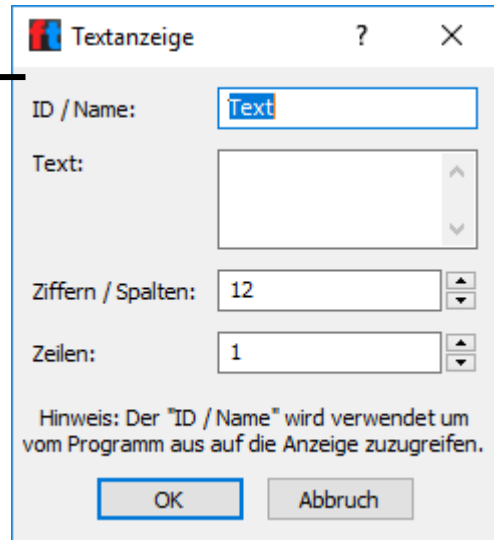
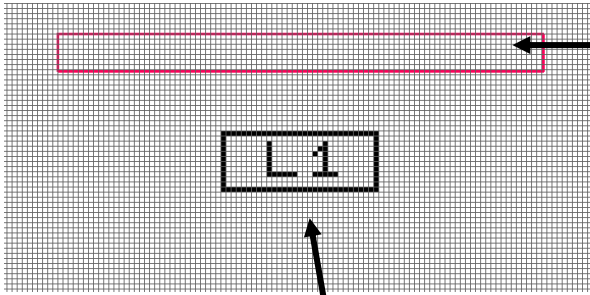


FTScratchTXT-Modul



Beispiel für einfache Programmierung

VIII Display und Taster TX-Controller

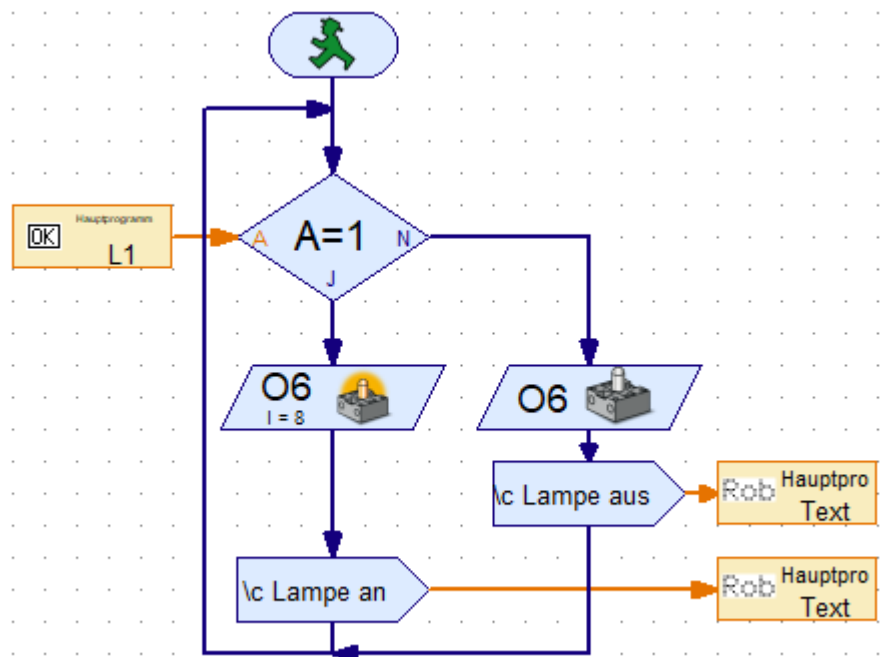


Reiter TXT/TX Display öffnen und eine Anzeige und ein Steuerelement einzeichnen.
 Kontextmenü (rechte Maustaste) auf Anzeige öffnet Menü für Textanzeige – Text eingeben für statischen Text oder offen lassen für dynamische Texte.
 Kontextmenü (rechte Maustaste) auf Steuerelement öffnet Menü Knopf. Beschriftung ändern und Werte für linken und rechten Knopf festlegen.

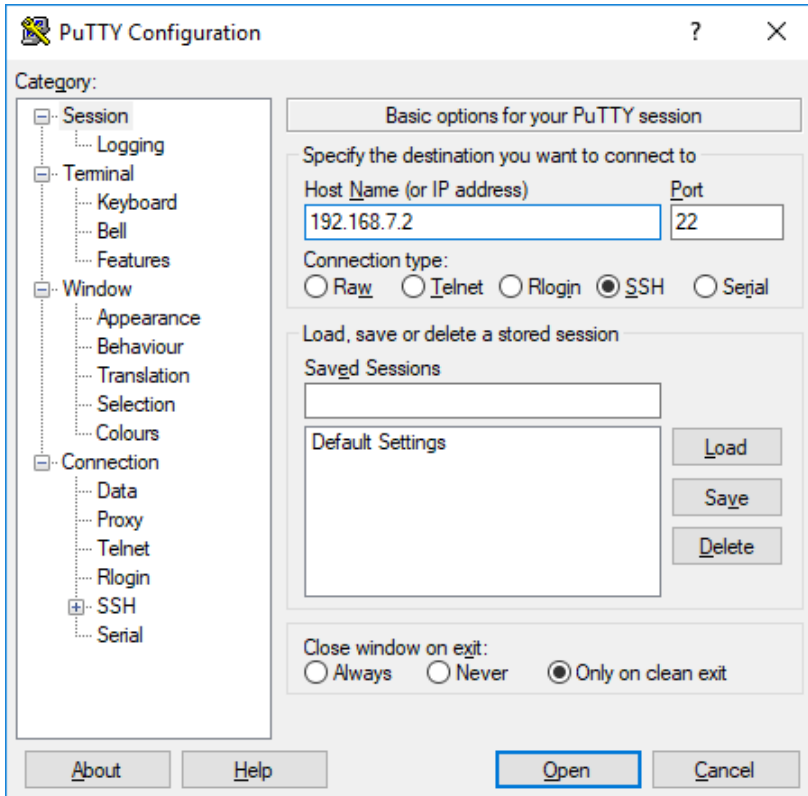


Programm:

Abfrage des Wertes Knopf L1. Ist Wert 1 – einschalten der Lampe O6 und Ausgabe des Textes „Lampe an“ Bedienfeldeingabe „Text“. Nach loslassen von Knopf 1 wird Wert1 wieder 0 – ausschalten der Lampe O6 und Ausgabe des Textes „Lampe aus“.



IX Zugriff auf Datenstruktur des TXT mit FTP



Putty.exe – Freeware laden, installieren und starten.

Folgende Eintragungen vornehmen:

Host: 192.168.7.2

Port: 22

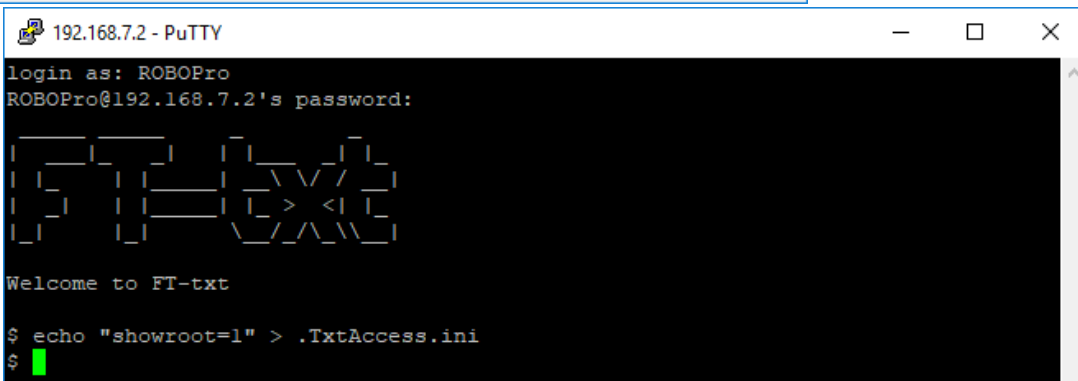
Connection: SSH

Close window on exit:

Only on clean exit

Button Open drücken

Neues Fenster erscheint



Login: ROBOPro

Passwort: ROBOPro

Eingabe: "echo showroot=1" > .TxtAccess.ini

Enter und Putty beenden

TXT neu starten



Einstellungen/ Info -> Root password wird angezeigt

FTP-Programm öffnen

FTP-Standort bearbeiten

Allgemein

Name:

Beschreibung:

Standort

Protokoll: SFTP (Secure FTP über SSH)

Server: Port:

StartDir ETP:

Passiver Modus

Benutzerinformation

Anmeldung:

Benutzername:

Kennwort:

Dateiname:

Name: TXT
 Protokoll: SFTP
 Server: 192.168.7.2
 Benutzer: root
 Kennwort: aus TXT ausgelesen

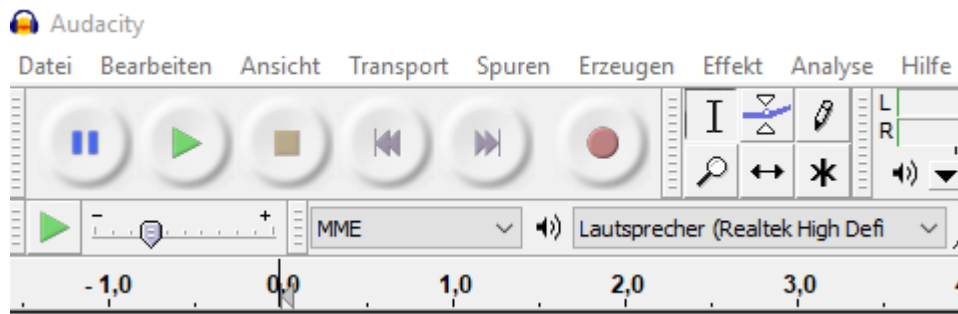
Nach dem öffnen dieser FTP-Verbindung, wird die Datenstruktur des TXT dargestellt:

Name	Größe	Typ	Geändert	Attr
bin		Dateiordner	01.01.2018 01:34	755
dev		Dateiordner	01.01.2018 01:00	755
etc		Dateiordner	01.01.2018 01:00	755
home		Dateiordner	27.06.2014 01:00	755
lib		Dateiordner	11.06.2016 01:00	755
lib32		Dateiordner	02.12.2014 01:00	777
libexec		Dateiordner	04.12.2014 01:00	755
linuxrc		Dateiordner	02.12.2014 01:00	777
media		Dateiordner	02.12.2014 01:00	755
mnt		Dateiordner	27.06.2014 01:00	755
opt		Dateiordner	30.06.2014 01:00	755
proc		Dateiordner	01.01.1970 01:00	555
root		Dateiordner	02.12.2014 01:00	700
run		Dateiordner	27.06.2014 01:00	777
sbin		Dateiordner	01.01.2018 01:34	755
sys		Dateiordner	01.01.2018 01:00	555
tmp		Dateiordner	01.01.2018 01:00	776
usr		Dateiordner	11.06.2016 01:00	755
var		Dateiordner	01.01.2018 01:05	755
libvncclient.so.0.0.0	0	0-Datei	26.12.2014 01:00	664
libvncserver.so.0.0.0	0	0-Datei	26.12.2014 01:00	664

X Änderung der Sounddateien

Wie in Punkt 9 die FTP-Verbindung öffnen und zu opt/knobloch/SoundFiles navigieren.

Diese Soundfiles können im Inhalt geändert werden, müssen aber im Namen genau der alten entsprechen.

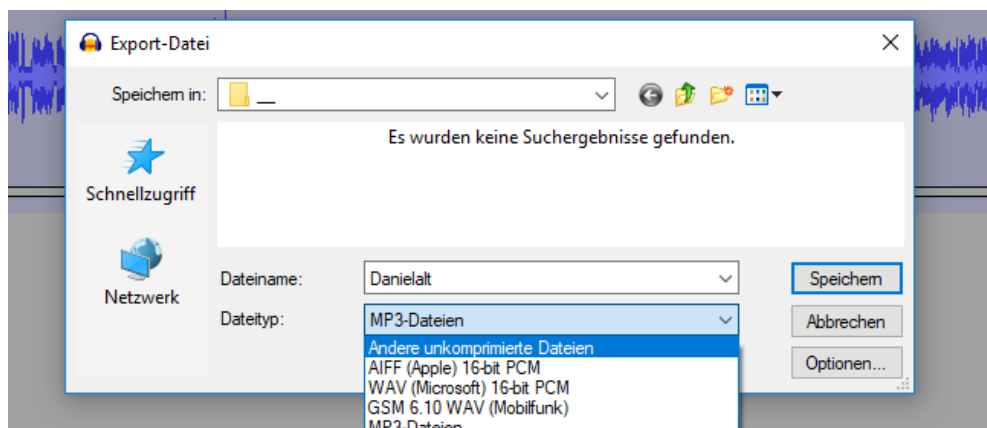
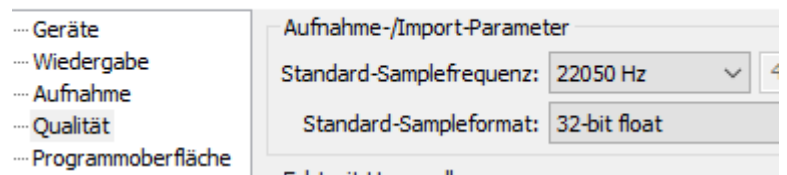


Ein neuer Sound kann z.B. mit Audacity erstellt werden.

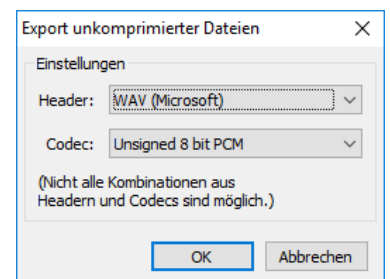
Als Voreinstellung unter Bearbeiten/ Einstellung ist die Frequenz und das Format entsprechend (linke Abb.) einzustellen.

Die Aufnahme muss in Mono erfolgen.

instellungen: Qualität



Abspeichern mit Datei/ Exportieren/ Dateityp – Andere unkomprimierte Dateien/ Optionen/ Header – WAV/ Code – unsigned 8 bit PCM/ Speichern



Ich habe vergeblich versucht, eine brauchbare einfache Anleitung im Netz zu finden. Auch das beiliegende Anleitungsbuch erschließt dem Anwender nicht alle Möglichkeiten. Deshalb habe ich mich in diesem Heft bemüht, alle Möglichkeiten zusammenzutragen. Die Themen sind dabei in der Reihenfolge nicht als Anleitung, sondern als Nachschlagewerk zu betrachten. Der erste Abschnitt wurde 2013 für das Robo Interface zusammen-gestellt. Der zweite Abschnitt wurde dann 2017 um die neuen Funktionen des TXT Controllers ergänzt. 2018 habe ich die Doku um Besonderheiten von Display und Tastern des TX-Controllers ergänzt.

Axel Chobe

