

I²C

fischertechnik

Zusammengestellt von Axel Chobe

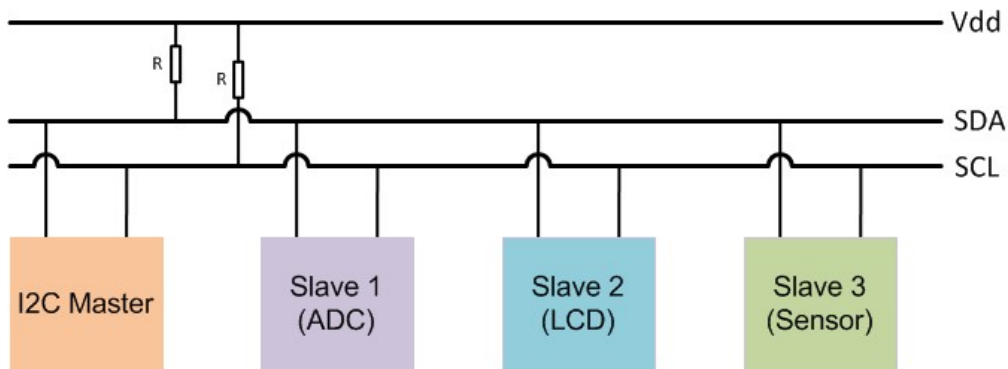
Aktuell 18.03.2023

Inhalt:

Grundlagen.....	3
Fischertechnik eigene Sensoren am TXT.....	4
Orientierungssensor 158402.....	4
Umweltsensor 167358.....	5
Gestensensor 186705	6
Initialisierungsmodul	6
Dämmerungsschaltermodul	6
Farbmodul HSV.....	6
Farbmodul RGB	7
Näherungsmodul	7
Gestensteuerungsmodul.....	7
Andere Sensoren am TX und TXT	8
Anschlüsse	8
Levelshifter	8
I2C Schreiben (Level 4) sendet Befehle oder Daten	9
I2C Lesen (Level 4) liest Daten von der Schnittstelle	9
IO Board PCF8574.....	10
Schaltkreis PCF8574.....	11
LED Display Sparkfun COM-11440	12
Variante 1 – ansprechen über Single Char.....	13
Variante 2 – ansprechen über Move Cursor	13
Real Time Clock (RTC) DS1307.....	15
Stellen der RTC	15
Auslesen der RTC.....	15
Nunchuck.....	17
Farbsensor TCS34725	18
LCD03-Display	20
Servo-Driver PCA9685	22
Mini-I²C-Servo-Adapter	23
OLED 128X64 Display SSD 1306.....	24

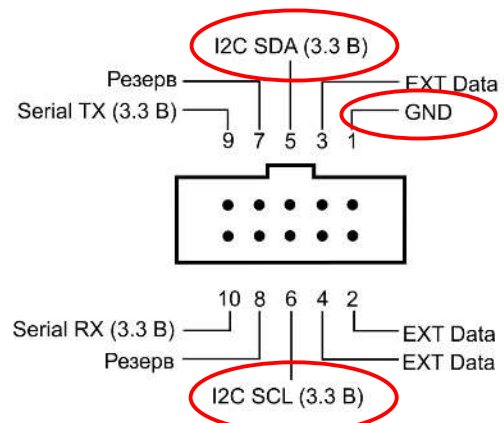
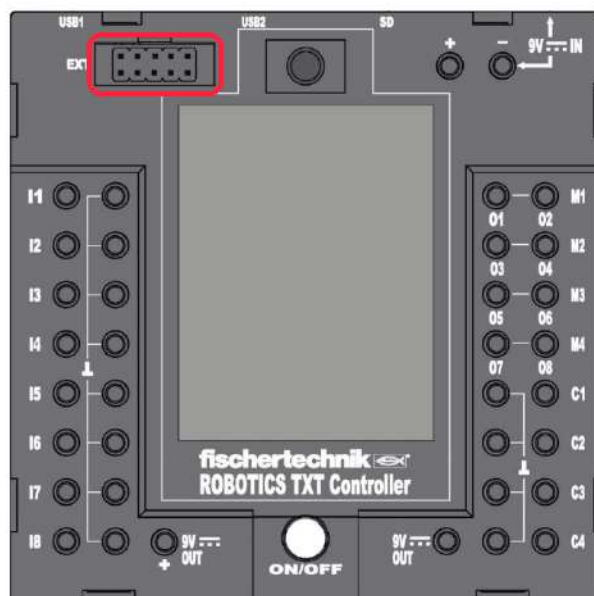
Grundlagen

Das I²C-Bus-System dient zum Aufbau und Betrieb von Geräten, für die die Anzahl der Steuerleitungen oder deren Belastbarkeit nicht ausreichen. Es handelt sich dabei um eine Steuermöglichkeit aus der Consumelektronik, die auf die Einsparung von Leitungen abzielt. Der I²C-Bus besteht aus 4 Leitungen, der +5V- Leitung und der Masseleitung sowie der Datenleitung SDA und der Taktleitung SCL. Diese verbinden einen Steuercomputer (PC oder Mikroprozessor) den so genannten Master mit einem oder mehreren Peripheriebausteinen, den Slaves. Hier sind Datenspeicher, I/O-Portbausteine, AD- oder DA-Wandler, Uhrenbausteine und diverse Anzeigentreiber möglich.



Das I²C-Protokoll wird auch von den fischertechnik-Controllern TX und TXT unterstützt. In der fischertechnik-Programmierungsumgebung ROBO Pro gibt es seit 2012 einen Lese- und einen Schreibbefehl, über den I²C-Sensoren angesprochen werden können. Der TX kann über die 5V-Stromversorgung des Busses problemlos Aktoren mit mehreren hundert mA Strombedarf versorgen.

Der TXT hingegen verwendet eine 3,3V-Logik und stellt an seinem Bus keine stabile Stromversorgung bereit. Mit einem *Level Shifter* und einem Spannungswandler (9V oder 5V auf 3,3V) lassen sich die Sensoren an beiden Controllern betreiben.

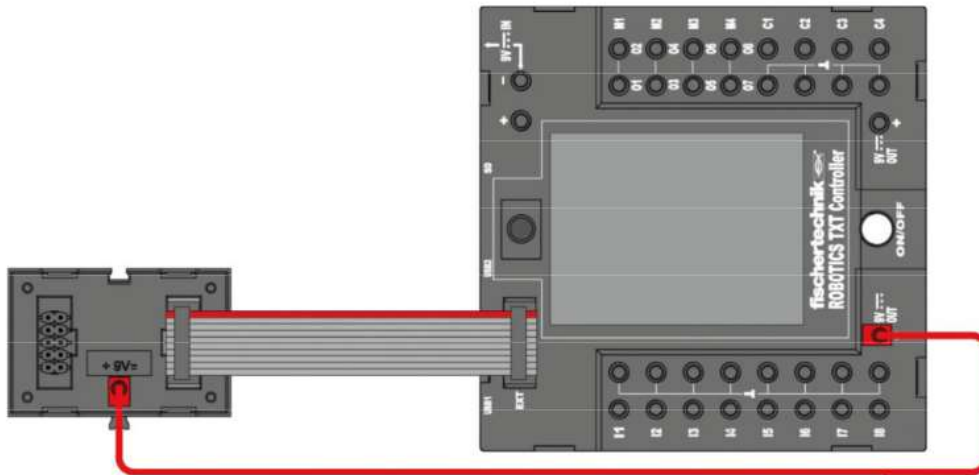


Fischertechnik eigene Sensoren am TXT

Orientierungssensor 158402



Der Fischertechnik 3-in-1 Orientierungssensor umfasst einen Beschleunigungssensor, ein Gyroskop und einen Kompasssensor, die über den I²C-Bus des ROBOTICS TXT Controllers als drei unabhängige Geräte angesprochen werden können. Die Verbindung des Kombisensors erfolgt über ein 10-poliges Flachbandkabel (117336), das an den Erweiterungseingang (EXT) des TXT Controllers angeschlossen wird sowie über ein einadriges Stromkabel (9V), das in den 9V OUT-Ausgang des TXT Controllers eingesteckt wird.



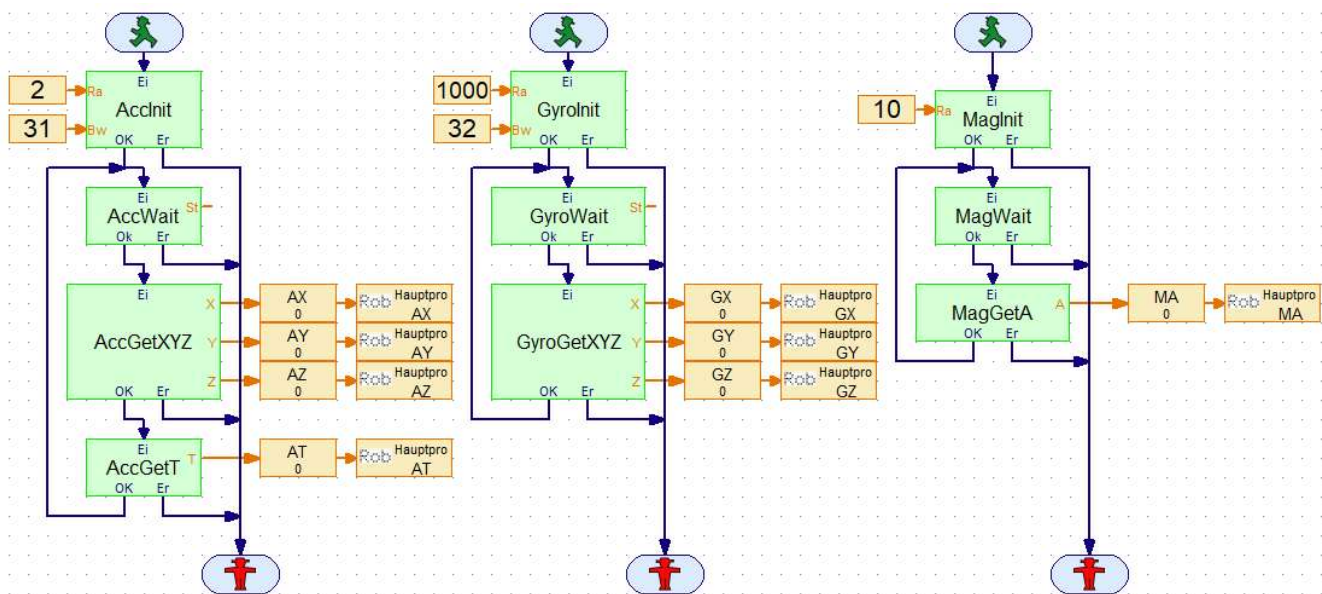
Spezifikationen:

- Integrierter 3-Achsen Gyrosensor / 3-Achsen Beschleunigungssensor
- Auflösung: bis zu 12 bit
- Messbereiche Beschleunigung: $\pm 2/\pm 4/\pm 8/\pm 16g$
- Messbereiche Gyroskop: $\pm 125/\pm 245/\pm 500/\pm 1000/\pm 2000^\circ/s$
- Abtastrate 1k/s
- I²C-Schnittstelle

I²C-Adresse der Sensoreinheiten

Beschleunigungssensor:	0x18
Gyrosensor:	0x68
Kompasssensor:	0x10

Eine typische Konfiguration ist das Beispielpogramm 'Acc-Gyro-BMX055.rpp'



Dieses Programm liest alle Werte des Sensors ein und zeigt sie im Downloadmodus auf dem Display des TXT.

Dieses Beispiel befindet sich im Ordner ROBOPRO\Bibliothek\I2C\Acc-Gyro-BMX055.rpp

Umweltsensor 167358



Dieser Sensor dient zur Messung von Luftdruck, Feuchte- und Lufttemperatur.

T Temperatur (°C)

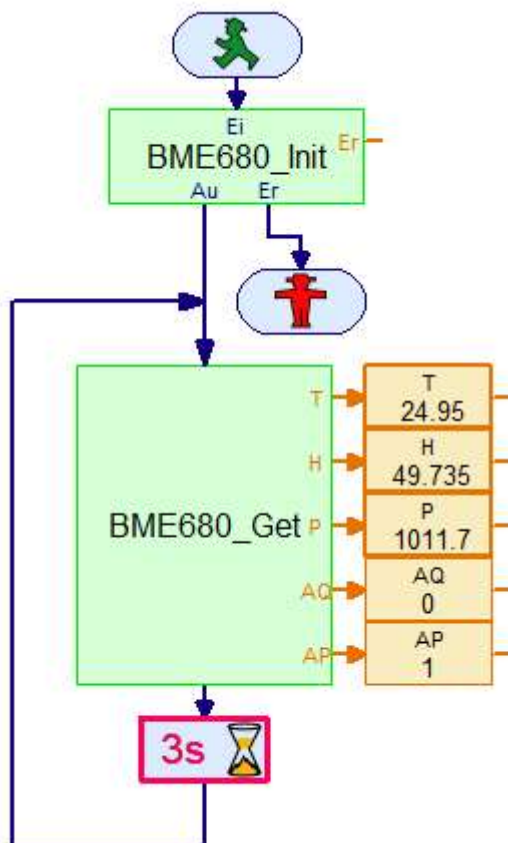
H Luftfeuchtigkeit (relative Luftfeuchtigkeit in %)

P Luftdruck (hPa für Hecto-Pascal)

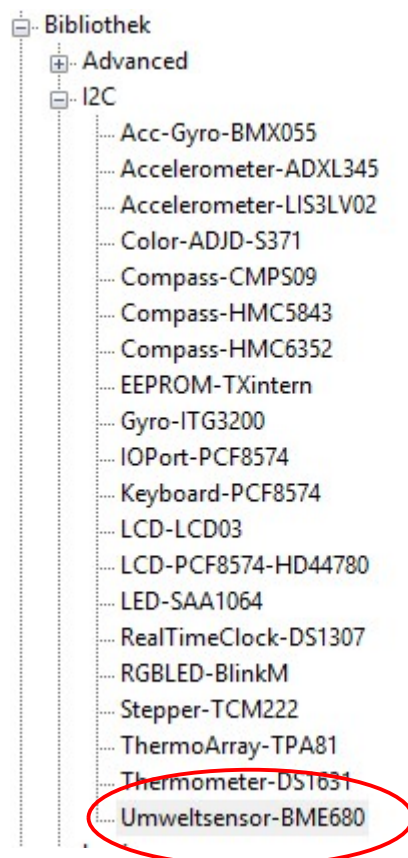
AQ Luftqualität (Messwert von 0 – 500) (0 = gut; 500 = schlecht)

AQP Verlässlichkeit der AQ-Werte bei der Messung über einen bestimmten Zeitraum (Messwerte 0-3) (0 = unbrauchbar; 3 = sehr verlässlich)

Dieses einfache Programm liest alle Werte des Sensors aus und gibt sie als Variablenwert (Gleitkomma) aus.
Weitere Beispiele unter Beispielprogramme\ TXT Smart Home \



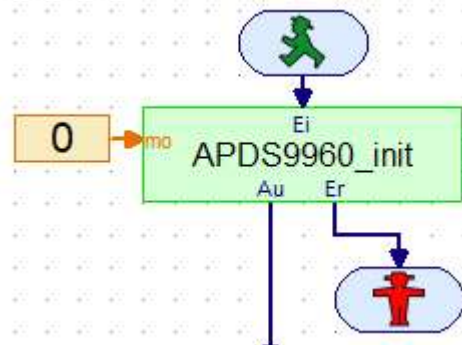
Die beiden Blöcke (BME680_Init und BME680_Get) befinden sich in der Elementgruppen Bibliothek:



Gestensensor 186705



0 Initialisierungsmodul



Vorbelegung

0 = alles deaktiviert

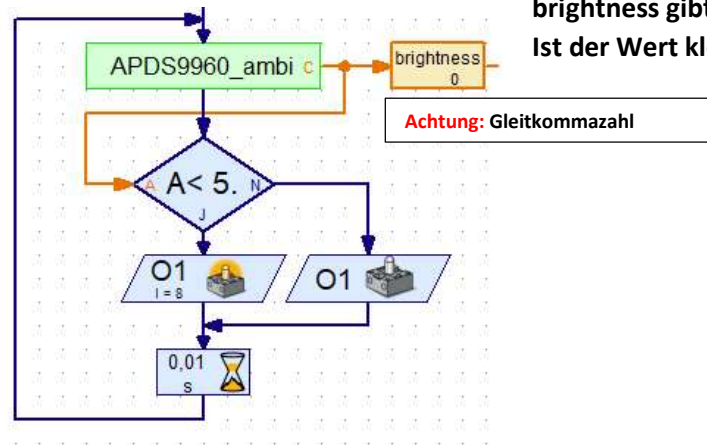
1 = Dämmerungsschalter, Farberkennung

2 = Näherungsaktivierung

3 = Licht und Nähe

4 = Gestensteuerung

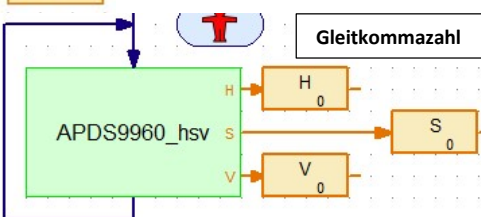
1 Dämmerungsschaltermodul



brightness gibt einen Wert zw. 0 und 100 aus.

Ist der Wert kleiner 5 geht das Licht an.

1 Farbmodul HSV

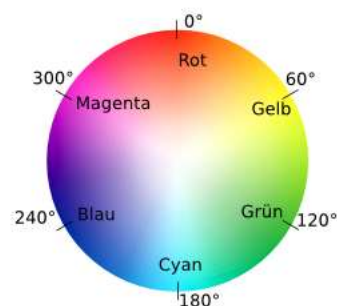


H = Farbwert (hue): Farbwinkel auf dem Farbkreis

S = Farbsättigung (saturation): (0% = Neutralgrau, 50% = wenig gesättigte Farbe, 100% = gesättigte, reine Farbe), Wert = 0 bis 1

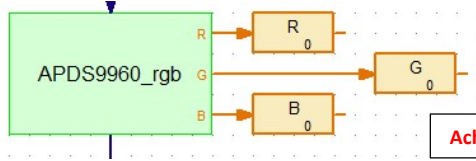
V = Hellwert (value; auch Dunkelstufe genannt): (0% = keine Helligkeit, 100% = volle Helligkeit), Wert = 0 bis 1

Das HSL-Farbmodell zeigt Farben anhand eines Kreises, in dem die Farben von außen nach innen immer weniger gesättigt sind. Auf dem Rand des Kreises liegen die Farbtöne (HUE) im Abstand zwischen 0 bis 360.



1

Farbmodul RGB

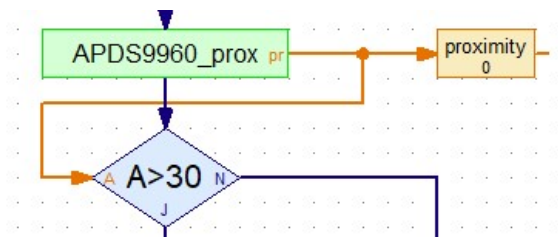


Ausgabe der Werte R, G und B im Bereich 0 bis 100

Achtung: Gleitkommazahl

2

Näherungsmodul

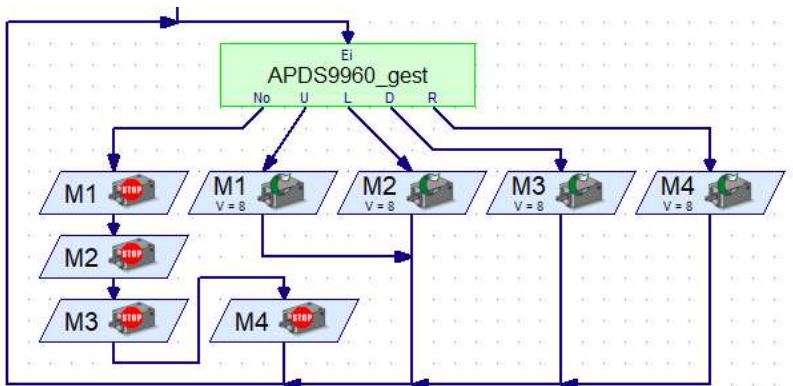


proximity gibt einen Wert zwischen 0 (weit) und 255 (nah) aus.

Ist der Wert größer 30 wird hier eine Aktion ausgeführt.

4

Gestensteuerungsmodul



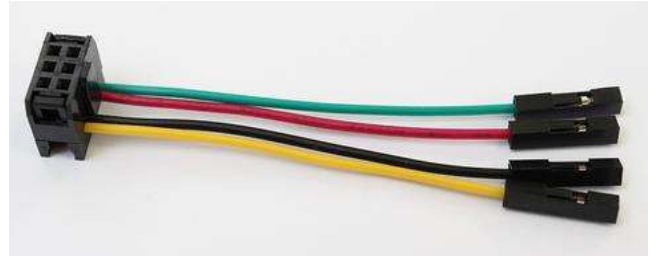
In Abhängigkeit der Richtung werden die Ausgänge U, L D und R aktiv.

Ansonsten wird Ausgang No aktiv.

Andere Sensoren am TX und TXT

Anschlüsse

Der TX kann über die 5V-Stromversorgung des Busses problemlos Aktoren mit mehreren hundert mA Strombedarf versorgen. Der Anschluss an die sechspolige EXT2-Buchse gelingt sehr einfach mit dem folgenden Adapter:



6	5
4	3
2	1

6: I²C Clock 5: I²C Data
4: RS485-B 3: RS485-A
2: 5 V DC Out 1: GND

10	9
8	7
6	5
4	3
2	1

10: Ser-RX (3V3) 9: Ser-TX (3V3)
8: Reserviert 7: Reserviert
6: I²C SCL Clock (3V3) 5: I²C SDA Data
4: Extension Data 3: Extension Data
2: Extension Data 1: Masse GND

Steckerbelegung I²C TX-Controller

Steckerbelegung I²C TXT-Controller

Levelshifter

Der TXT hingegen verwendet eine 3,3V-Logik und stellt an seinem Bus keine stabile Stromversorgung bereit. Somit besteht beim Anschluss eines I²C-Devices immer die gleiche Frage nach Stromversorgung und Logikspannung. Mit sogenannten Levelshiftern lassen sich Devices mit unterschiedlichen Logikspannungen miteinander verbinden. Der ft-Extender löst beide Probleme, indem er einen Levelshifter und zwei Linearregler

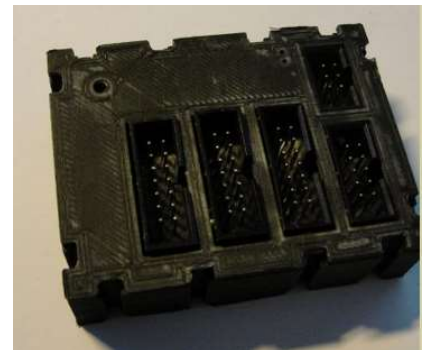
auf der Platine hat. Somit können problemlos 5V- und 3,3 V-Devices ohne Zusatzbeschaltung über ein

Der 6-polige Stecker ist kompatibel zum EXT-Anschluss des TX und ftDuinos, so dass hier die beiden Geräte direkt angeschlossen werden können. Für den Anschluss reicht ein 1:1 belegtes 6-poliges Kabel.

GND	1	2
SDA 5V	3	4
	5	6

Der 10-polige Anschluss ist kompatibel zum TXT. Hier kann direkt der TXT oder vorhandene Sensoren mit 10-poligem Stecker angeschlossen werden. Für den Anschluss reicht ein 1:1 belegtes 10-poliges Kabel.

GND	1	2
SDA 3,3V	3	4
	5	6
	7	8
	9	10



einfaches Kabel angeschlossen werden. Als Stromversorgung stehen 250 mA für jedes angeschlossene Device zur Verfügung. Dies ist für den Betrieb von Sensoren völlig ausreichend.

Die Betriebsbereitschaft wird über zwei LEDs angezeigt.

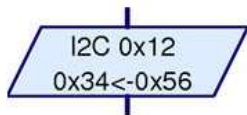
An der 9 V-Buchse wird die ft-Spannung angeschlossen, um die Versorgung der Levelshifter und Sensoren zu ermöglichen. Es ist nur die Buchse für +9V Gleichspannung vorhanden, der Minuspol ist auf jedem Flachbandkabel zu dem TX, TXT oder ftDuino bereits vorhanden.

Die drei 14-poligen Stecker sind für den kombinierten Einsatz von 3.3 V- und 5 V-Sensoren gedacht. Dabei sind die Pins 1-6 in der Belegung mit dem TXT, die Pins 9-14 sind mit dem TX/ftDuino kompatibel. An Pin 7 und 8 stehen die Versorgungsspannungen 3,3 V und 5 V zur Verfügung.

GND	1	2
SDA 3,3V	3	4
3,3V	5	6
GND	7	8
	9	10
SDA 5V	11	12
	13	14

Der ft-Extender und weitere Informationen unter:
<https://gundermann-software.de/shop/>

I2C Schreiben (Level 4) sendet Befehle oder Daten



The screenshot shows the "I2C Schreiben" dialog box. It contains the following fields and options:

- Geräteadresse: 0x0
- Unteradresse: 0x0
- Dateneingang: ☐
- Datenwert: 0x0
- Länge der Unteradresse: 8 bit
- Länge der Daten: 8 bit
- Geschwindigkeit: 400kHz
- Fehlerbehandlung: Wiederholen bis erfolgreich
- Offen lassen: ☐
- Buttons: OK, Abbruch

Das Element *I2C Schreiben* sendet ein Adressbyte und zwischen 1 und 4 Datenbytes über die I2C Schnittstelle. Zunächst werden die 7-bit Geräte-adresse und das Schreibbit gesendet. Anschließend wird eine 0-2 Bytes lange Unteradresse und zuletzt 1 oder 2 Datenbytes gesendet. Aus der Sicht des I2C Protokolls besteht kein Unterschied zwischen Unteradresse und Daten.

Geräteadresse: 7 Bit Geräteadresse (ohne R/W Bit) Bei manchen Geräten ist die Adresse als 8 Bit Adresse (mit R/W Bit) angegeben. In diesem Fall die Adresse durch 2 teilen. Die Geräte-adressen 0x50..0x57 (=0xA0...0xAF als 8 bit) werden vom TX Controller intern verwendet und können nicht für externe Bausteine verwendet werden.

Unteradresse: 8 oder 16 Bit Unteradresse (siehe Länge)

Dateneingang: Auswahl, ob die Daten über Dateneingang oder Datenwert verwendet werden.

Datenwert: für Datenwert wenn kein Eingang

Länge der Unteradresse: Auswahl, ob kein, 8 oder 16 bit. Bei 16 bit wird unterschieden ob das höher-wertige (MSB=Most Significant Byte) oder nieder-wertige (LSB=Least Significant Byte) Byte zuerst übertragen wird.

Länge der Daten: 8 oder 16 Bit Daten an Baustein

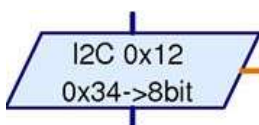
Geschwindigkeit: I²C-Takt 100kHz oder 400kHz

Fehlerbehandlung: Wiederholen oder sofort abbrechen (es erscheint ein weiterer Ausgang)

Offen lassen: gesetzt – das Element sendet kein Stopp.

Dadurch können weitere Elemente geschrieben oder gelesen werden.

I2C Lesen (Level 4) liest Daten von der Schnittstelle



The screenshot shows the "I2C Lesen" dialog box. It contains the following fields and options:

- Geräteadresse: 0x0
- Unteradresse: 0x0
- Länge der Unteradresse: 8 bit
- Länge der Daten: 8 bit
- Geschwindigkeit: 400kHz
- Fehlerbehandlung: Wiederholen bis erfolgreich
- Offen lassen: ☐
- Buttons: OK, Abbruch

Geräteadresse: 7-Bit Geräteadresse

Unteradresse: 8 oder 16 Bit Unteradresse

Länge der Unteradresse: siehe schreiben

Länge der Daten: 8 oder 16 Bit Daten vom Baustein

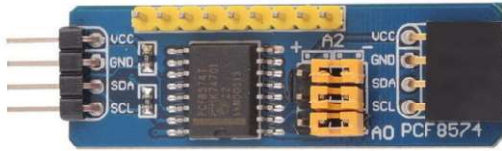
Geschwindigkeit: I²C-Takt 100kHz od. 400kHz

Fehlerbehandlung: Wiederholen oder sofort abbrechen (es erscheint ein weiterer Ausgang)

Wenn eine Unteradresse verwendet wird, sendet das Element *I2C Lesen* zunächst ein Adressbyte im Schreibmodus und anschließend die 1 oder 2 Byte lange Unteradresse. Anschließend führt das Element einen Restart am I2C Bus durch, sendet erneut die Geräteadresse, diesmal im Lesemodus, und liest anschließend 1-2 Datenbytes. Wenn keine Unter-adresse verwendet wird, wird das Adressbyte gleich im Lesemodus gesendet und anschließend die Daten gelesen.

IO Board PCF8574

Für TX und TXT mit Levelshifter !



Mit diesem Modul kann man IO Ports über einen I2C Port erweitern. Die I2C-Adresse kann über die Jumper eingestellt werden. Somit können 8 Adressen gewählt und auch 8 Boards an einem I2C betreiben werden. Insgesamt lassen sich somit 64

zusätzliche I/O-Anschlüsse erzeugen. Die Boards lassen sich dabei einfach hintereinander gesteckt kaskadieren. Die Betriebs-spannung liegt zwischen 2,5 und 6 Volt. Damit ist die Erweiterung für den **TX** und **TXT**-Controller geeignet.

Adresstabelle

A0	A1	A2	Adresse
0	0	0	0x20
1	0	0	0x21
0	1	0	0x22
1	1	0	0x23
0	0	1	0x24
1	0	1	0x25
0	1	1	0x26
1	1	1	0x27



Adressierung der I/O

P7	P6	P5	P4	P3	P2	P1	P0	INT
128	64	32	16	8	4	2	1	

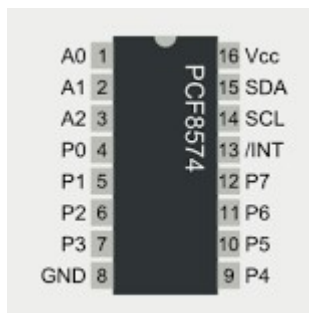


Durch Addition der Werte können auch mehrere I/O gleichzeitig angesprochen werden. (z.B. 33 für Port 0 und Port 5)

Beispiel zum Einlesen von Daten	Beispiel zum Ausgeben von Daten
<div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> Einstellungen Kontext Geräteadresse 0x20 Unteradresse 0x0 Länge UA keiner Länge Daten 8 bit Geschwindigk. 100 KHz Fehlerbehandl. Wiederh. Offen lassen leer </div>	<div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> Einstellungen Kontext Geräteadresse 0x20 Unteradresse 0x00 Dateneingang leer Datenwert 8 bzw. 0 Länge UA keiner Länge Daten 8 bit Geschwindk. 100 KHz Fehlerbehandl. Wiederh. Offen lassen leer </div>
Entsprechend der Verbindung von VCC zu den Eingängen 0 bis 7 wird in der Variable der entsprechende Binärwert ausgegeben (1, 2, 4, 8, 16, 32, 64, 128)	An den Ausgang von P4 (Binärwert 16) wird im Rhythmus von 1 Sekunden eine Spannung von 5 Volt gegen VCC ausgegeben. (z.B. für LED)

Schaltkreis PCF8574

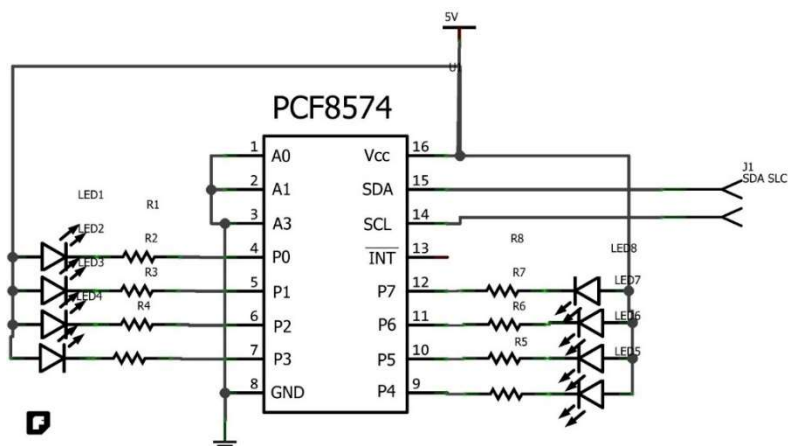
Für TX und TXT mit Levelshifter !



Der PCF8574 ist ein sogenannter I/O Port-Expander. Man bekommt in ca. für 1 €. An weiteren Bauteilen braucht man nur ein paar Widerstände. Der PCF8574 hat 8 Pins welche man als Ein- oder Ausgänge nutzen kann. Die Kommunikation mit dem Controller erfolgt über den I2C Bus. Der PCF8574 ist eine einfache und günstige Möglichkeit den I2C Bus auszuprobieren.

Hinweis:

Zum Ermitteln der aktuellen I²C-Adresse gibt es das Programm
i2cidentification_v15.rpp



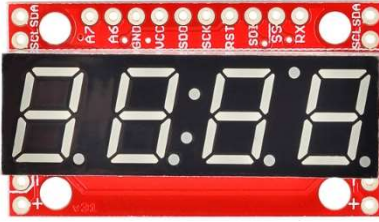
A0	A1	A2	8574 AN	8574 N
0	0	0	0x38	0x20
1	0	0	0x39	0x21
	1	0	0x3A	0x22
1	1	0	0x3B	0x23
0	0	1	0x3C	0x24
1	0	1	0x3D	0x25
0	1	1	0x3E	0x26
1	1	1	0x3F	0x27

Der PCF8574 hat drei Adresspins (A0..A2). Damit lassen sich maximal 8 Schaltkreise an einem gemeinsamen Bus betreiben. Da jeder über 8 Eingänge /Ausgänge verfügt könnte man damit zum Beispiel 64 LED's steuern. Im einfachsten Fall setzt man alle Adresspins auf Masse und der Schaltkreis hat die Adresse 38 Hex für 8574 AN. Für den Schaltkreis 8574 N gilt die Adresse 20 Hex.

Beispiel zum Einlesen von Daten	Beispiel zum Ausgeben von Daten
<p>Entsprechend der Verbindung von VCC zu den Eingängen 0 bis 7 wird in der Variable der entsprechende Binärwert ausgegeben (1, 2, 4, 8, 16, 32, 64, 128) Im Beispiel VCC an Pin 5)</p>	<p>An den Ausgang von P3 (Binärwert 08) wird im Rhythmus von 0,5 Sekunden eine Spannung von 5 Volt ausgegeben. (z.B. für LED) Bei größeren Lasten sollte eine externe Stromversorgung über einen Stepp-Down Converter benutzt werden.</p>

LED Display Sparkfun COM-11440

Für TX und TXT mit Levelshifter !



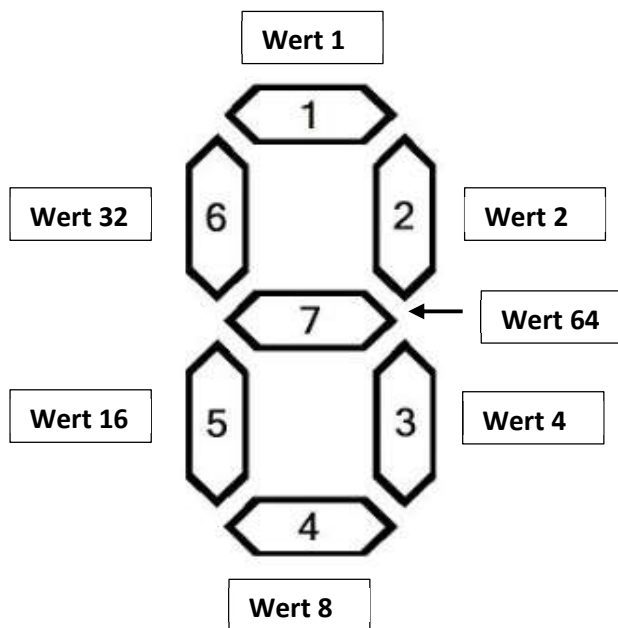
Das Sparkfun-Display ist eine vierstellige 7-Segmente-Anzeige, die mit **2,4 bis 5,5 V** betrieben wird. Der Strom-verbrauch beträgt bei 5,5 V max. 14,1 mA und bei 3,3 V nur 7,9 mA. Damit kann das Display am **TX** als auch am **TXT** betrieben werden. Die I²C-Adresse ist mit **0x71** vor-konfiguriert, welche aber per Software geändert werden kann (max. 126 Adressen). Die I²C-Anschlüsse sind gekoppelt und so angeordnet, dass sich mehrere Module in Reihe

schalten lassen.

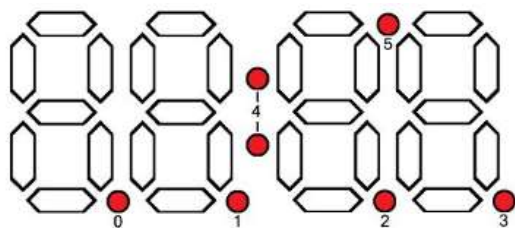
Befehle

Wert	Kommando	Parameter
0x76	Clear Display	-
0x77	Decimal Points	0x00 – 0x3F (siehe Anhang)
0x79	Move Cursor	{0, 1, 2, 3}
0x7A	Brightness	{0, ..., 255} *1
0x7B	Single Char 1 (li.Segment)	0x00 – 0x7F (siehe Anhang)
0x7C	Single Char 2	0x00 – 0x7F
0x7D	Single Char	0x00 – 0x7F
0x7E	Single Char	0x00 – 0x7F
0x7F	Baud Rate	0x00 – 0x0B
0x80	Set I ² C-Adress	0x01 – 0x7E
0x81	Reset (Adresse auf 0x71)	-

***1 Die Werte 117 und 118 (0x76 und 0x77) besitzen einen Firmenware-Bug und sollten nicht an das Display gesendet werden.**



Zur Darstellung werden die Werte der 7 Seg. addiert.



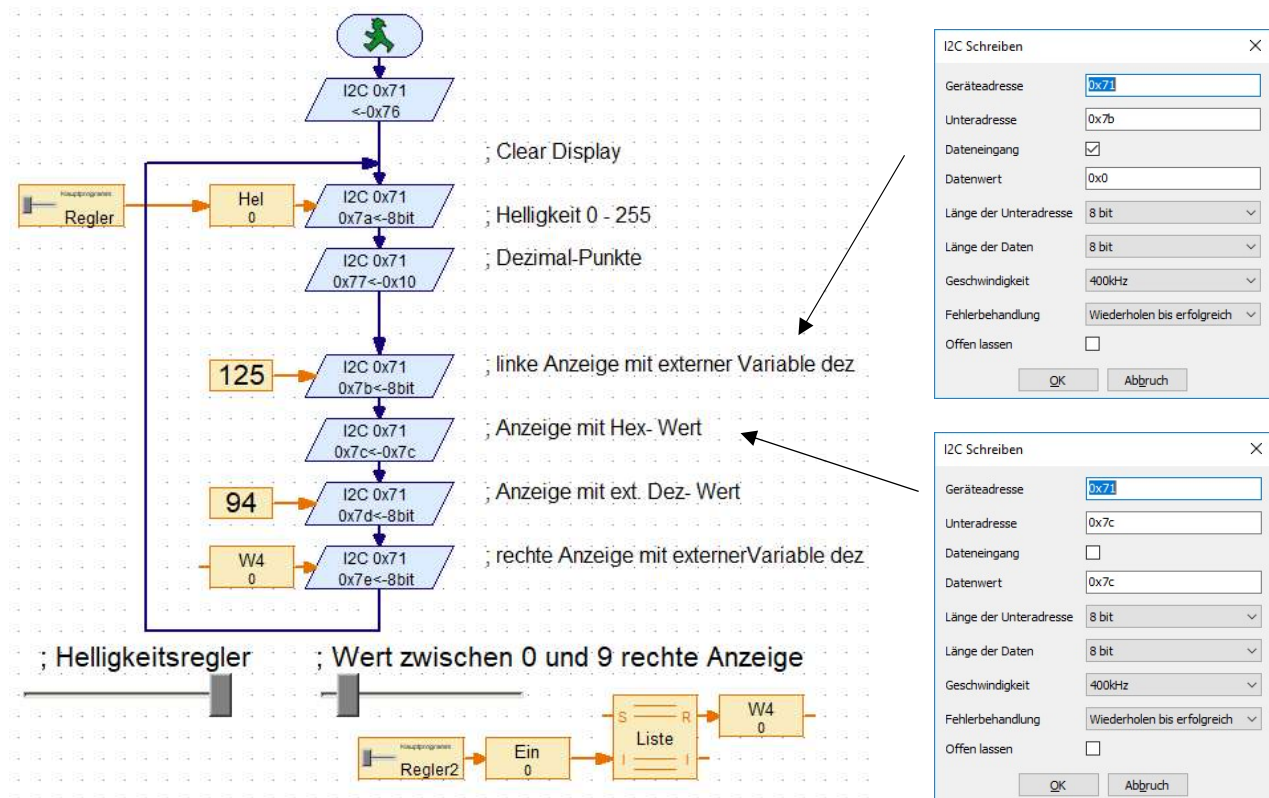
Durch Addition der Werte können auch mehrere Dezimalpunkte gesetzt werden.

Ziffer	Wert Dez	Wert Hex
1	6	6
2	91	5B
3	79	4F
4	102	66
5	109	6D
6	125	7D
7	7	7
8	127	7F
9	111	6F
0	63	3F
A	119	77
B	124	7C
C	57	39
D	94	5E
E	121	79
F	113	71

Pos.	Wert Dez	Wert Hex
0	1	1
1	2	2
2	4	3
3	8	8
4	16	10
5	32	20

Variante 1 – ansprechen über Single Char

Mit den Kommandos 0x7B bis 0x7E können die LED-Segmente einzeln angesprochen werden.



Den ersten 3 Segmente sind im Beispiel Werte vorgegeben. Das rechte Segment zeigt Werte zwischen 0 bis 9. Die Werte können über den Regler eingegeben werden. Mit dem Wert des Reglers (0 bis 9) wird ein Wert in der Liste angesprochen, der wiederum an die Variable W4 ausgegeben wird. Über den anderen Regler kann die Helligkeit zwischen 0 und 255 gesteuert werden.

Name:

Maximalgröße:

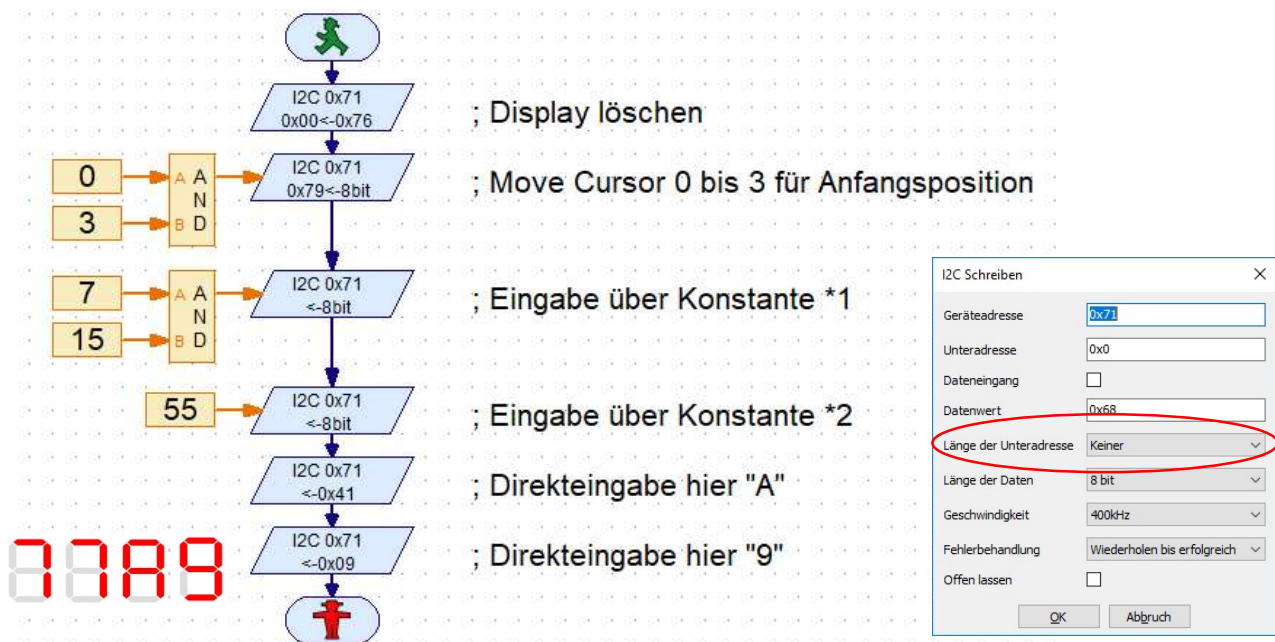
Anfangsgröße:

Liste der Anfangswerte:

63
6
91
79
102
109
124
7
127
111

Variante 2 – ansprechen über Move Cursor

Der Befehl 0x79 gefolgt von einem Bytewert aus {0, 1, 2, 3} setzt den Positionszeiger auf eines der vier 7-Segment-Anzeigen. Wenn anzeigbare Daten empfangen werden, wird das Zeichen an der aktuellen Cursorposition angezeigt, und der Cursor bewegt sich dann um einen Punkt vorwärts. Wenn ein Zeichen auf der vierten Stelle (ganz rechts) angezeigt wird, springt der Cursor zur ersten Stelle (ganz links).



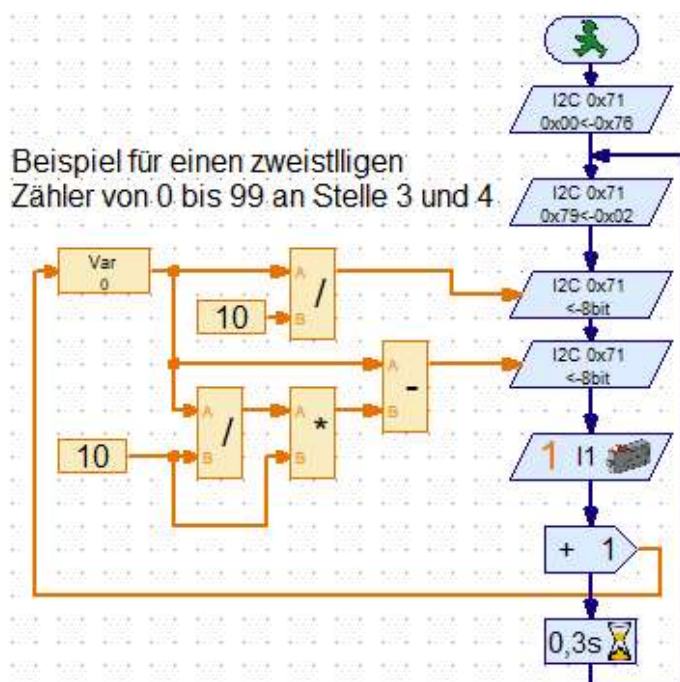
*1 So können nur die Ziffern 0 bis 9 dargestellt werden. Zur Fehlervermeidung UND-Verknüpfung mit 15 zur Darstellung der letzten 4 Bits

*2 Bin. Darstellung von 55 = 0011 0111; Darstellung der letzten 4 Bits - Anzeige 7

Über die direkte Eingabe im I²C-Element können auch weitere Zeichen eingegeben werden.

Zeichen	Byte	Zeichen	Byte	Zeichen	Byte	Zeichen	Byte	Zeichen	Byte
0	0x00	A	0x41	g	0x67	p	0x70	-	0x2D
1	0x01	B	0x42	H	0x48	Q	0x51	„	0x22
2	0x02	C	0x43	h	0x68	q	0x71	`	0x27
3	0x03	C	0x63	J	0x4A	R	0x52	_	0x5F
4	0x04	D	0x46	j	0x6A	r	0x72	Leer	0x78
5	0x05	E	0x45	L	0x4C	T	0x54		
6	0x06	e	0x65	N	0x4E	t	0x74		
7	0x07	F	0x46	n	0x6E	U	0x55		
8	0x08	f	0x66	o	0x6F	u	0x75		
9	0x09	G	0x47	P	0x50	Y	0x59		

Beispiel für einen zweistelligen
Zähler von 0 bis 99 an Stelle 3 und 4

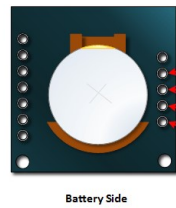
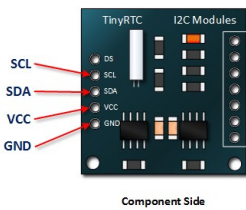


Der Tastendruck wird in der Variablen gespeichert. Die Zehneranzeige erfolgt durch Division mit 10 und die Eineranzeige durch Errechnung des Modulo.

Real Time Clock (RTC) DS1307

Für TX und TXT mit Levelshifter !

Für zahlreiche Anwendungen, die mit Zeitangaben arbeiten ist eine exakte Uhr hilfreich. Der Timer von Robo



Pro ist dazu zu ungenau, kennt kein Datum und keine Uhrzeit und vergisst alles nach dem ausschalten. Daher sollte die Echtzeituhr über eine unabhängige Stromversorgung, wie das RTC1307 verfügen.

Die I²C-Adresse lässt sich nicht ändern und ist fest auf **0x68** eingestellt. Der DS1307 beherrscht lediglich zwei Befehle: einen

Schreib- und einen Lese-Modus, in denen Uhrzeit und Datum eingestellt bzw. ausgelesen werden.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 4	Bit 2	Bit 1	Bit 0
68x00	10 Sekunden				Sekunden			
68x01	10 Minuten				Minuten			
68x02	0	12/24	A/P	10	Stunden			
68x03	0	0	0	0	0	Wochentag		
68x04	0	0	10 Tag		Tag			
68x05	0	0	0	10 M	Monat			
68x06	10 Jahr				Jahr			
68x07	OUT	0	0	SQWE	0	0	RS1	RS0

Ein sehr einfacher Robo Pro-Treiber für den DS1307 steht in der aktuellen Robo Pro-Version bereit. Die Module befinden sich Bibliothek / I²C / RealTimeClock-DS1307.

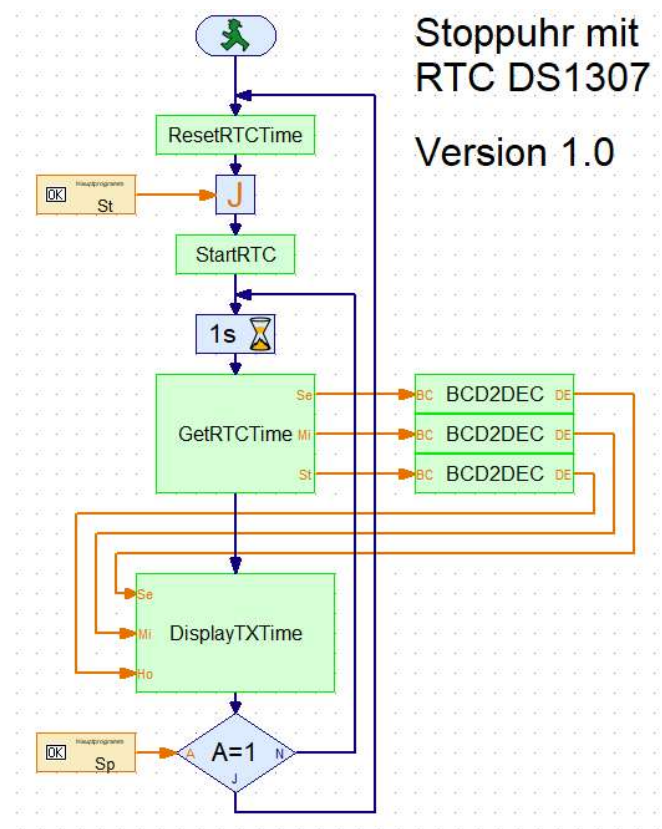
Stellen der RTC

Zum Stellen der Uhr wird Byte für Byte ein Write-Befehl an die Slave Adresse 0x68 Geschickt, gefolgt von der Registeradresse, in die geschrieben werden soll (Sub-adresse) und dem BCD-kodierten zugehörigen Bytewert das Uhrzeit- bzw. Datumselement.

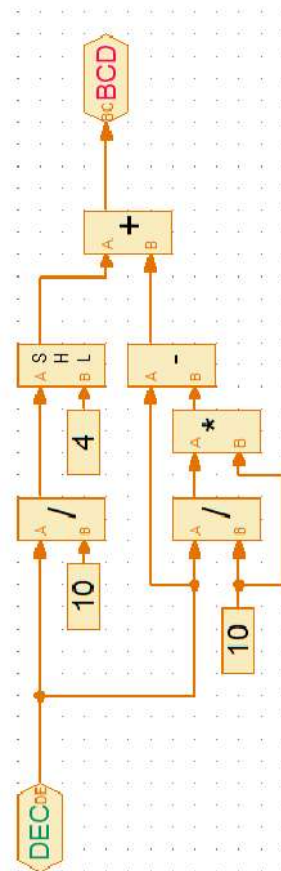
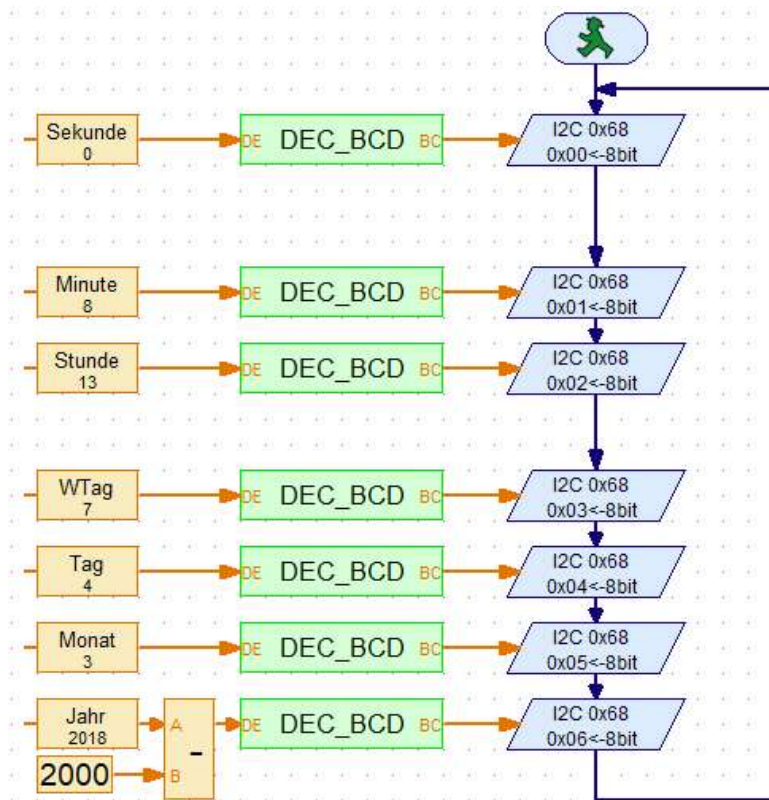
Das Clock Halt-Bit (CH) im Sekunden-Byte schaltet den Oszillator ab. (z.B. um den RTC als Stoppuhr zu verwenden siehe Beispiel)

Auslesen der RTC

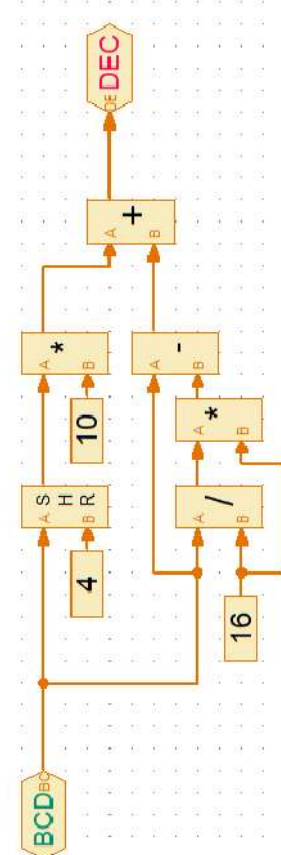
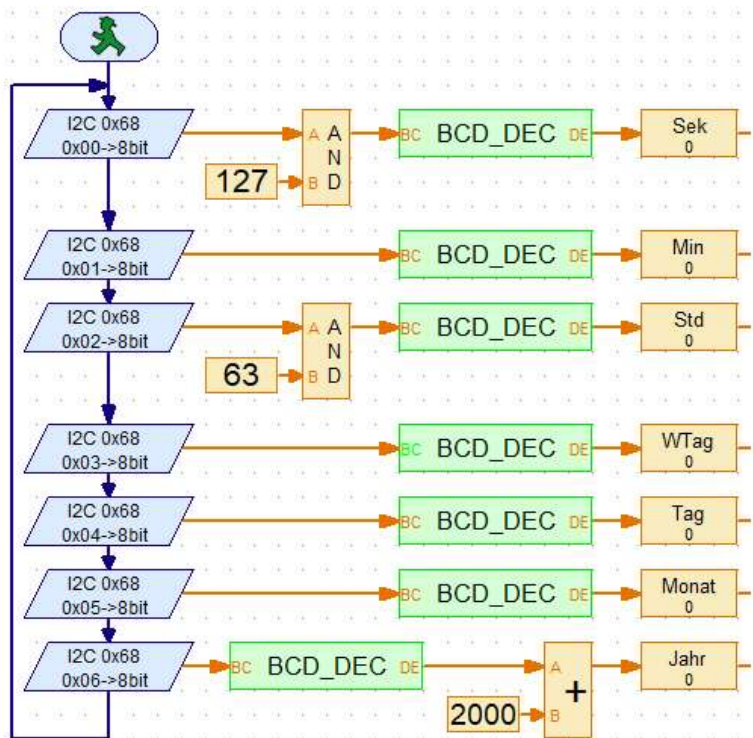
Das Auslesen der Zeit erfolgt mit dem Read-Befehl an die Slave-Adresse 0x68, an den sich die Angabe der auszulesenden Register-Adresse anschließt. So können die BCD-Werte Byte für Byte ausgelesen werden.



Beispiel für Eingabe aller Werte (inkl. Unterprogramm DEC_BCD)

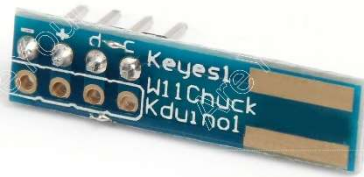


Beispiel für Ausgabe aller Werte (inkl. Unterprogramm BCD_DEC)



Nunchuck

Für TX und TXT mit Levelshifter !



Die Videospielkonsole Wii ermöglicht durch Verwendung einer Infrarotkamera und Beschleunigungssensoren dreidimensionale Spielerlebnisse. Dazu gibt es den an den Controller ansteckbaren Nunchuck mit einem analogen Joystick, zwei Tastern und einem 3-Achs-Beschleunigungs-sensor. Die Kommunikation erfolgt über das I²C Protokoll. Die technischen Daten sind: **3,3**

Volt Stromversorgung; feste **Adresse 0x52**; Joystick mit zwei 30 K-Potentiometer – zwei digitale 8-bit-Werte von 0 bis 255.

Bit	7	6	5	4	3	2	1	0
Byte 1	Joystick (X-Achse, 0 - 255)							
Byte 2	Joystick (Y-Achse, 0 - 255)							
Byte 3	Beschleunigung (X, Bits 9 -2)							
Byte 4	Beschleunigung (Y, Bits 9 -2)							
Byte 5	Beschleunigung (Z, Bits 9 -2)							
Byte 6	Z 1/0	Y1/0	X 1/0	C	Z			

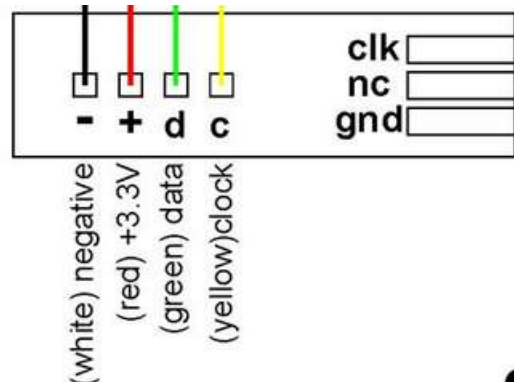
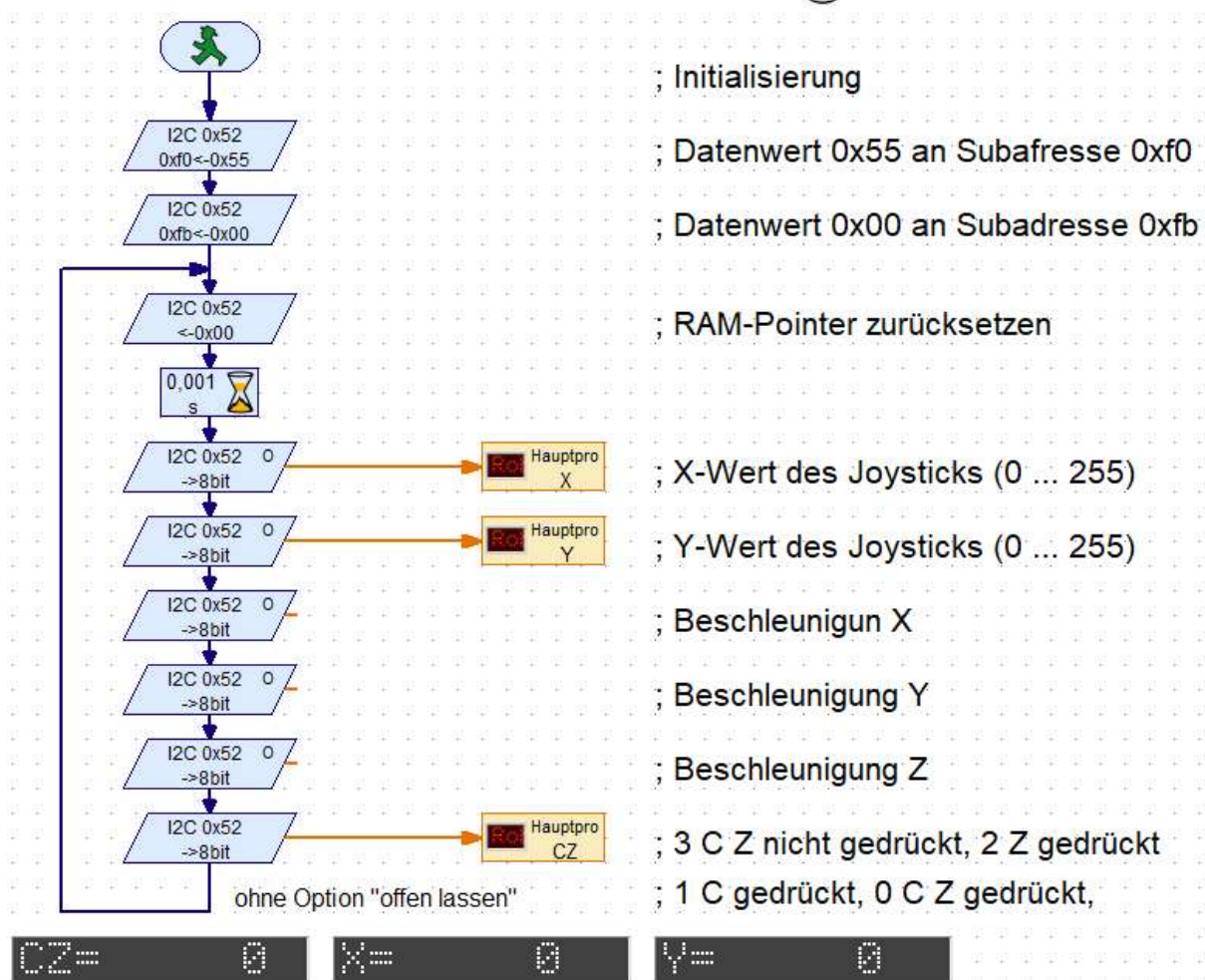


Bild rechts: Anschluss Belegung des Steckadapters für den Nunchuck



Farbsensor TCS34725

Für TX und TXT mit Levelshifter !



Für die Bestimmung von Farben gibt es von FT einen Farbsensor. Er verfügt über eine rote LED, deren Licht von verschiedenfarbigen Flächen unterschiedlich stark reflektiert wird. Er erkennt also nicht die Farbe, sondern die Helligkeit der Reflexion. Das Ergebnis ist allerdings stark abhängig vom Abstand und vom Umgebungslicht. Eine Alternative sind RGBC-Farbsensoren, die die Farbe in die Farbbestandteile Rot, Grün, Blau und Weiß zerlegen und deren Anteil bestimmen. Sie sind genauer und unabhängig von Umwelteinflüssen.

Was ist Licht?

Als Licht wird das Spektrum der für Menschen sichtbaren Wellenlänge elektromagnetischer Strahlung bezeichnet. Es umfasst die Wellenlänge von 380 nm (violett) bis 780 nm (dunkelrot). Weißes Licht ist eine Mischung der Frequenzen aus dem gesamten Spektrum.

Was sind Farben?

Farbe entsteht, wenn man nur ein Teil des Lichtspektrums sieht. Ein Objekt erscheint farbig, wenn seine Oberfläche nur einen ausgewählten Frequenzbereich des sichtbaren Lichtes reflektiert. Der Sensor misst also die Frequenzbereiche des vom Objekt reflektierten Lichtes.

TCS34725

Die Fotodioden zur Farbbestimmung sind mit einem IR-Filter ausgestattet. Die Platine besitzt außerdem eine LED, mit der das Objekt beleuchtet wird. Die technischen Daten sind:

Betriebsspannung: 3,3-5 Volt (um den Sensor mit beiden Spannungen nutzen zu können sollte der Anschluss Vin statt 3V3 benutzt werden). Stromaufnahme: 0,235 mA; I²C-Adresse: fest auf **0x29**; I²C-Geschwindigkeit: 400 kbit/s. Der LED-Sensor hat eine Farbtemperatur von 4150 K. Wenn nur die Helligkeit oder Farbtemperatur gemessen werden soll, kann die LED abgeschaltet werden (Kontakt LED mit GND-Anschluss verbinden)

Ein **Command-(Byte-)Befehl** ist an einem gesetzten Bit7 (0x80) erkennbar, gefolgt von der Selektion des Lese- oder Schreibmodus in Bit 5 und 6. In den Bits 0-4 stehen die Adressen des Registers, das nachfolgend (zuerst) gelesen bzw. beschrieben werden soll.

7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1
1	0	0	1	0	1	0	0

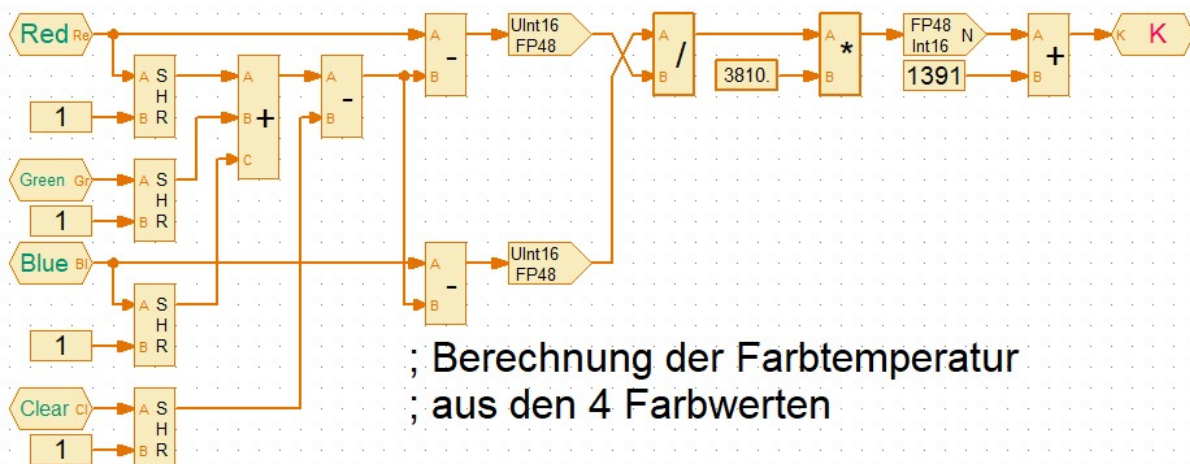
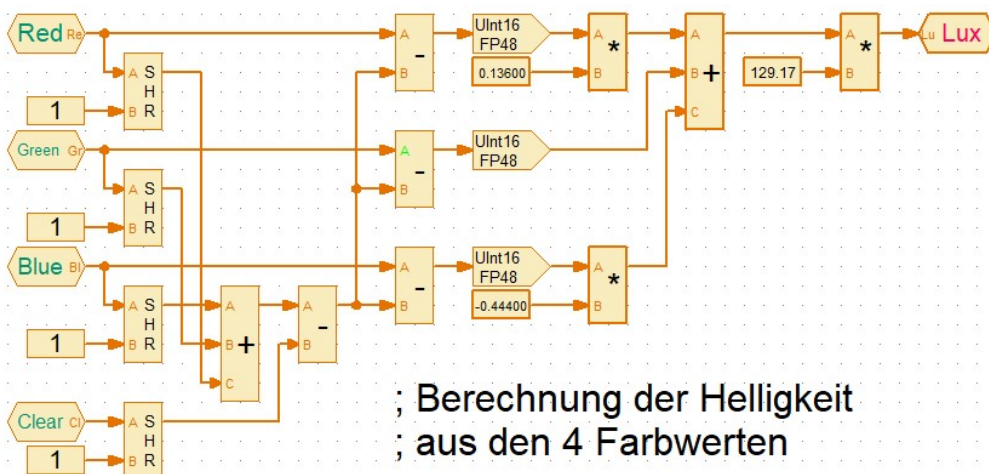
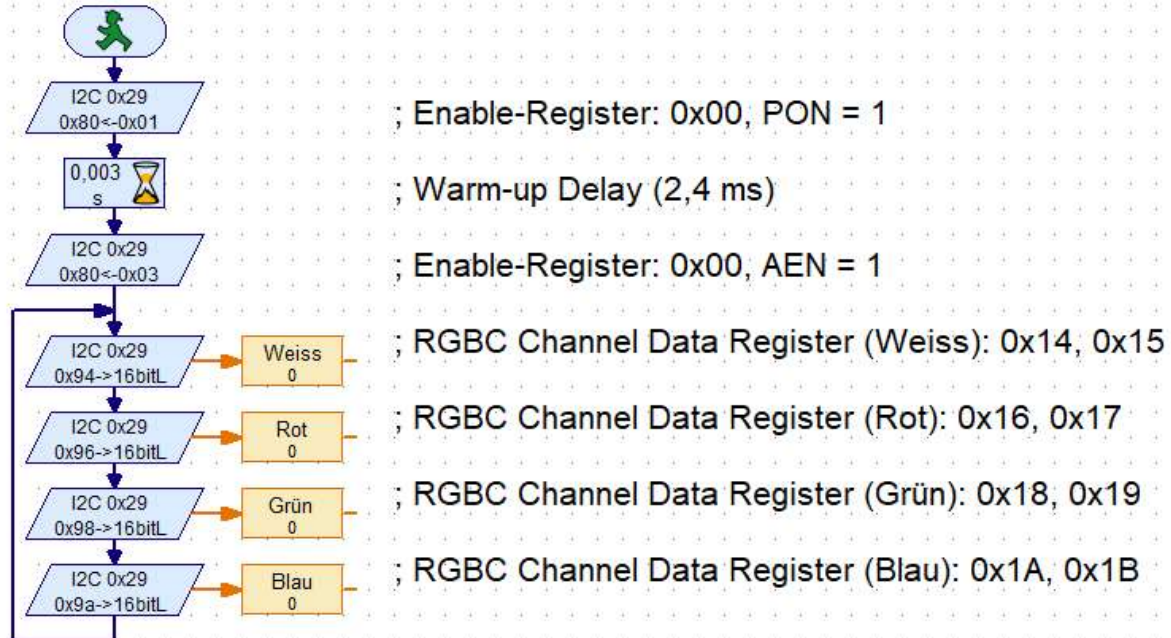
128 Dez bzw. 80 Hex für Command-Befehl

Beispiel für wecken des Sensors (Bit 0 auf 1)

Beispiel für lesen weiss (80 Hex + 14 Hex = 94 Hex)

Register	Bedeutung	Erläuterung
0x00	Enable Register	Nach dem Einschalten ist der Sensor im SleepZustand. Bit 0 auf 1: (PON) der Sensor wird geweckt Bit 1 auf 1: (AEN) ADC-Enable aktiviert den A/D-Wandler Bit 3 auf 1: Aktiviert die Energiesparfunktion (WEN) Bit 4 auf 1: (AIEN) Signalisierung ein Interrupt aktiviert
0x01	RGBC Timing Register	Kann die Integrationszeit der A/D Wandlung festlegen. Default ist 0xFF (ein Integrationszyklus)
0x03	Wait Time Register	Länge der Wartezeit vor einer Messung. Erforderlich, wenn die Energiesparfunktion im Enable Register gesetzt ist.
0x04-07	Interrupt Threshold	Auslösen eines Interrupts, wenn ein Schwellwert des Weiss-Sensors über oder unterschritten wird.
0x0C	Persistence Register	Einstellung der Zahl der Messwerte die in Folge für die Auslösung des Interrupts.
0x0D	Configuration Register	Default 0x00, ansonsten 0x01 um die Zeit im Wait Time Register um den Faktor 12 zu verlängern.
0x0F	Control Register	Verstärkungsfaktor der Fotodiode (00=einfach, 01= 4fach; 10=16fach; 11=60fache Verstärkung)

0x12	ID Register	Hier ist der Wert 0x44 (=68) fest eingetragen. So kann eine Prüfung erfolgen, ob der Sensor richtig angeschlossen ist.
0x13	Status Register	Zeigt an, wenn die Farbbestimmung abgeschlossen ist. Bit 0 gesetzt = Berechnung abgeschlossen. (siehe Beispiel)
0x14-1B	RGBC Channel Date	Die gemessenen Werte werden als vier 16-bit-Werte bereitgestellt. Sie verteilen sich wie folgt: Weiß: 0x14 – 0x15 ; Rot: 0x16-0x17; Grün: 0x18-0x19; Blau: 0x1A-0x1B



LCD03-Display

Für TX und TXT mit Levelshifter !



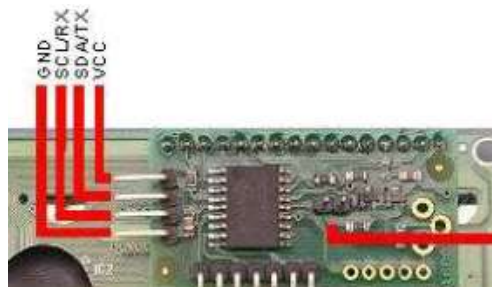
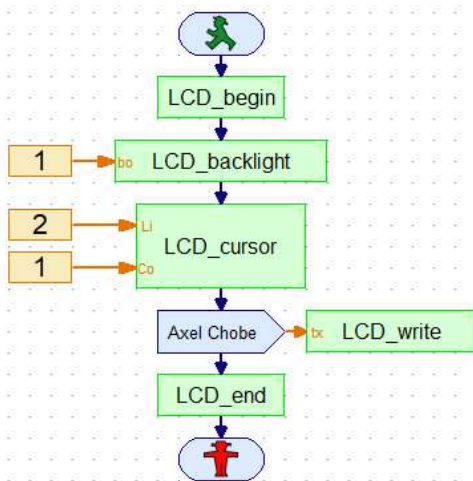
Das Display hat 2 Zeilen a 16 Zeichen. Es hat die feste Adresse 0x63 und wird mit 5 Volt betrieben. Die dazugehörigen Bibliotheken sind bereits in RoboPro enthalten. Die Helligkeit der LED-Hintergrundbeleuchtung kann mit einem auf der Platine angebrachten Poti mit einem kleinen Schraubenzieher manuell eingestellt werden.

Links ein Beispiel für einfache Textausgabe.

Daneben die Beschaltung vom I²C-Bus:

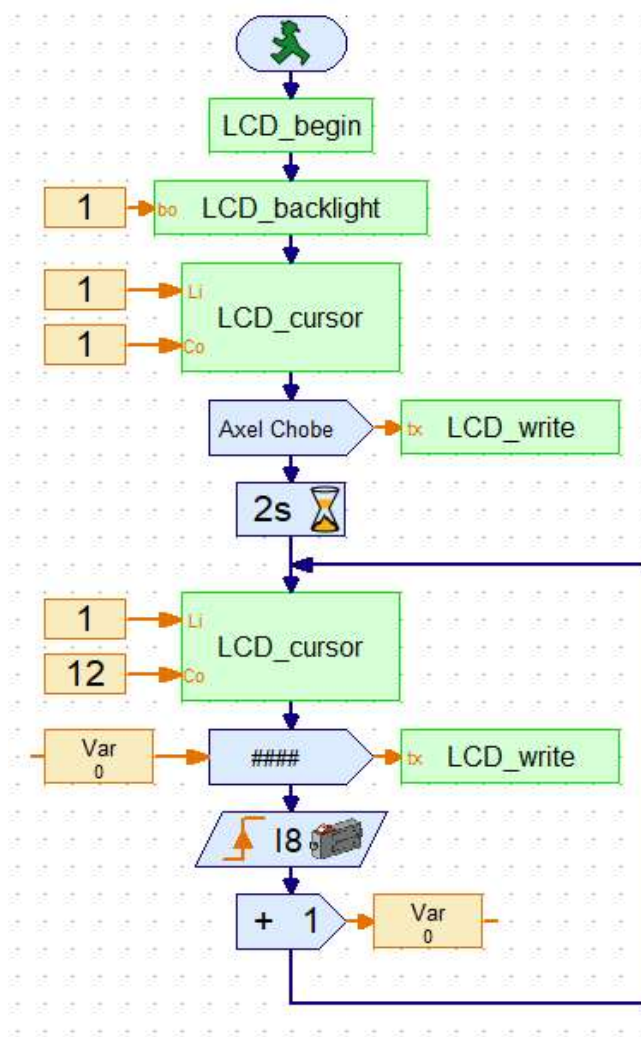
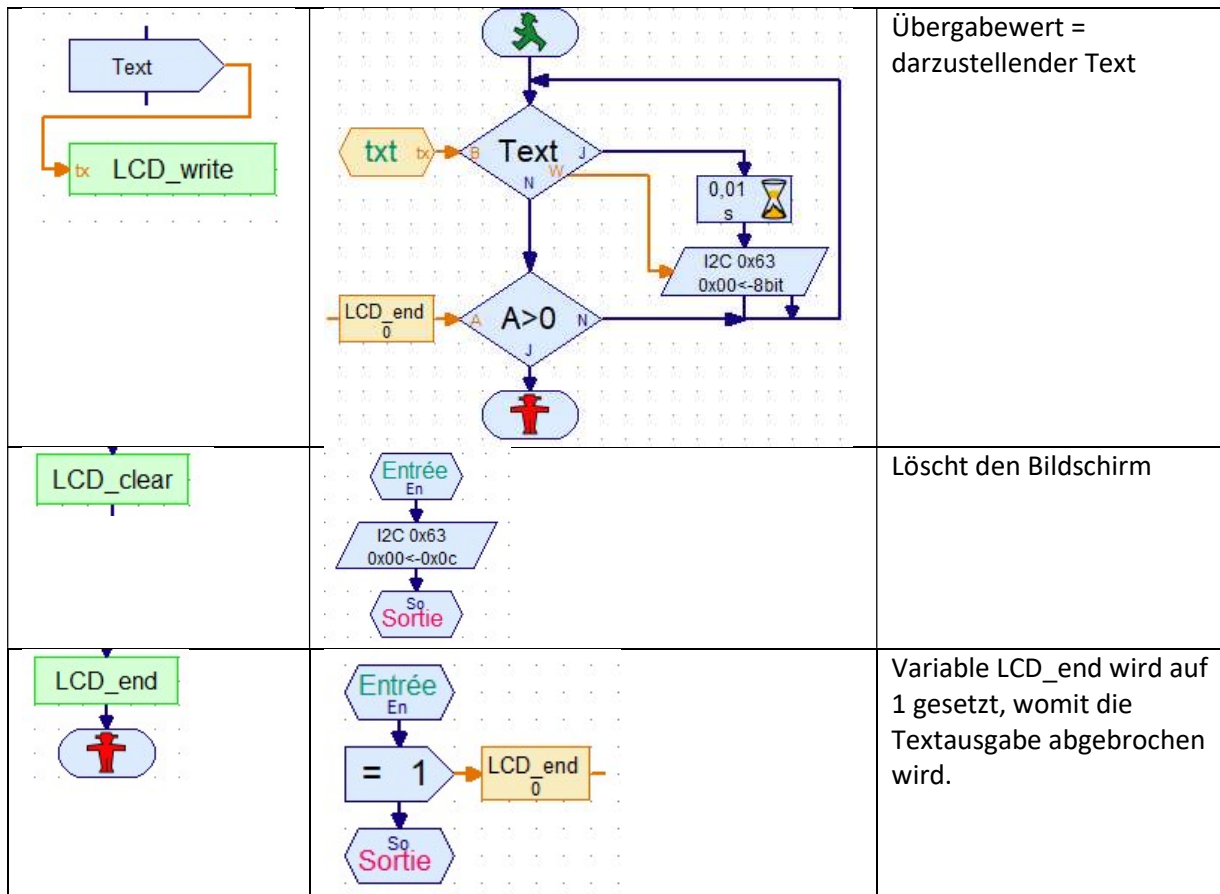
GND | SCL | SDA | VCC

Rechts der Hinweis auf Jumpermode (I²C ohne Jumper)



Mode jumper
open = I2C control
closed = Serial control

	<pre> graph TD En[Entrée En] --> I2C1[/I2C 0x63 0x00<-0x0c/] I2C1 --> I2C2[/I2C 0x63 0x00<-0x01/] I2C2 --> I2C3[/I2C 0x63 0x00<-0x04/] I2C3 --> So[Sortie] </pre> <p>clear</p> <p>home</p> <p>no cursor</p>	<p>LCD_begin Löschen des Bildschirms Cursor an erste Position Cursor nicht anzeigen</p>
	<pre> graph TD En[Entrée En] --> A{A>0} bo[boolean bo] --> A A -- N --> I2C1[/I2C 0x63 0x00<-0x14/] A -- J --> I2C2[/I2C 0x63 0x00<-0x13/] I2C1 --> So[Sortie] I2C2 --> So </pre> <p>backlight on</p> <p>backlight off</p>	<p>Übergabewert 1 = an Übergabewert 0 = aus</p>
	<pre> graph TD En[Entrée En] --> I2C1[/I2C 0x63 0x00<-0x03/] Li[Line Li] --> I2C2[/I2C 0x63 <-8bit/] Co[Column Co] --> I2C3[/I2C 0x63 <-8bit/] I2C1 --> So[Sortie] I2C2 --> So I2C3 --> So </pre>	<p>Übergabewert 2 = Zeile Übergabewert 1 = Spalte</p>



Beispiel für eine Textausgabe und einem Zähler über Taster 8. Als Platzhalter werden im Textanzeige Element Rautezeichen eingetragen.

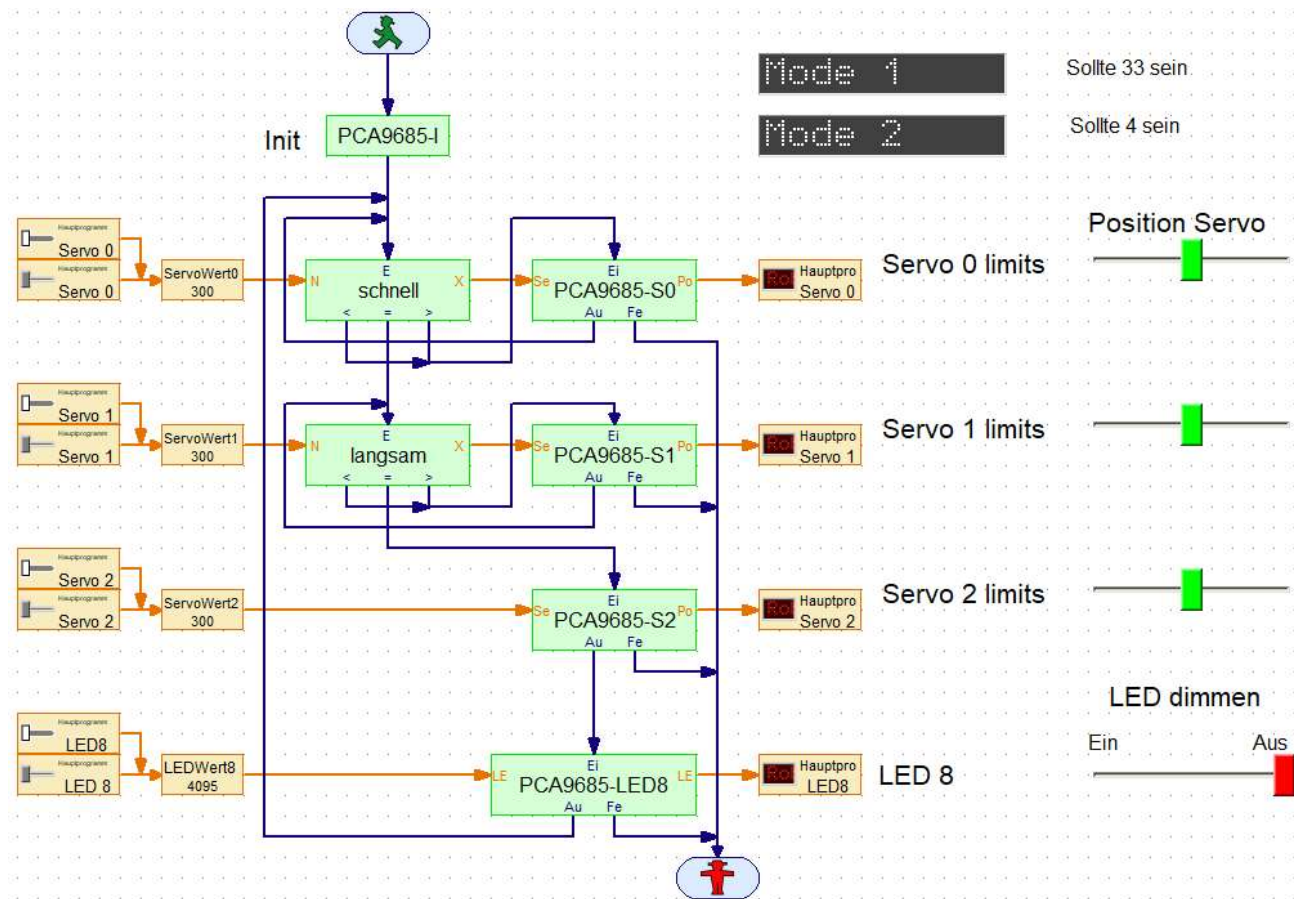
Servo-Driver PCA9685



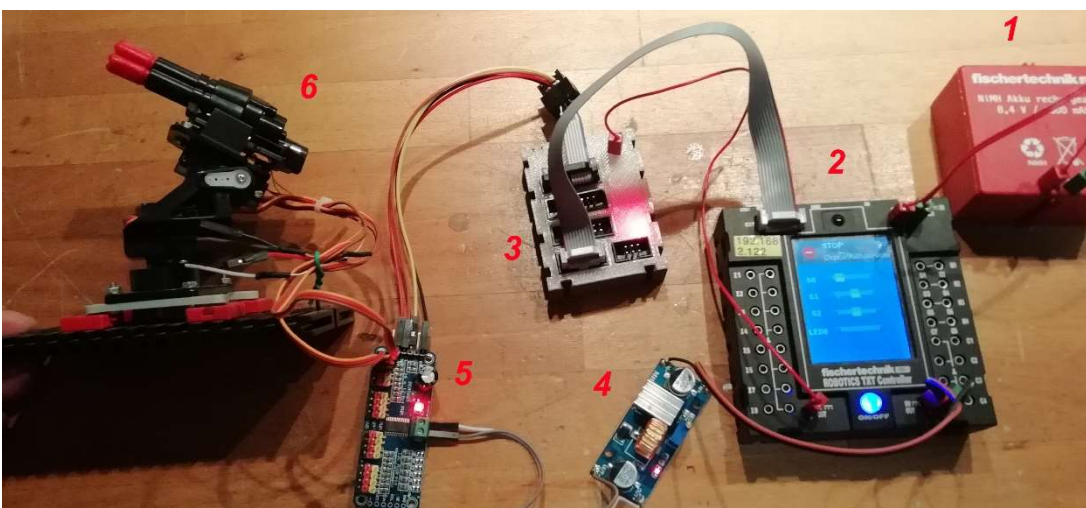
Die Fischertechnik-Controller TX und TXT können keine Servo-Motoren ansprechen. Immer wenn eine schnelle und Winkelgenaue Motorbewegung benötigt wird, ist ein Sensor einem ausgewachsenen Elektromotor vorzuziehen. Servo-Motoren sind 5V-Elektromotoren mit einer Steuerelektronik, die über ein pulswidenmoduliertes (PWM)-Signal gesteuert wird.

Folgendes Programm gibt es unter der Adresse:

<http://ftcommunity.de/data/downloads/robopro/i2cpca9685servoansteuerungversion2.0.rpp>



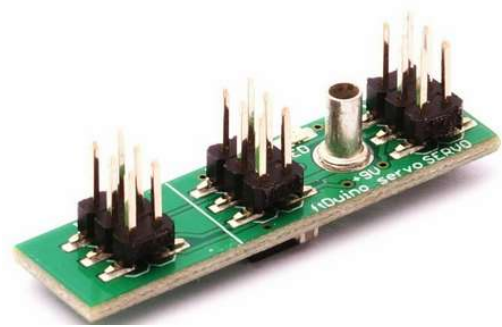
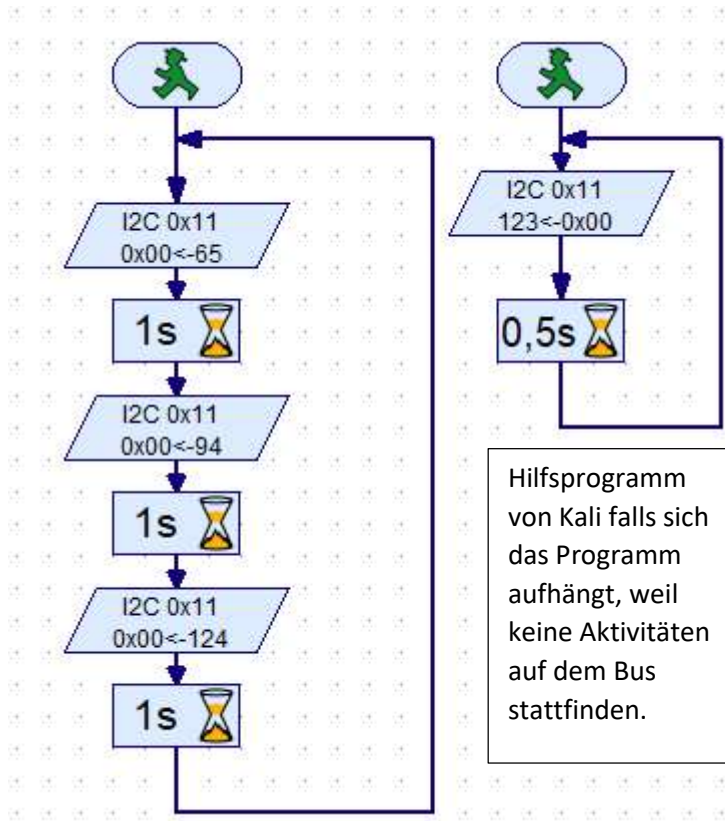
Die Geschwindigkeit kann eingestellt werden indem man ein Unterprogramm vorschaltet (schnell, langsam). Ein neues UP extra für LEDs. LED kann aus-, eingeschaltet oder gedimmt werden. Die Mittelstellung der Servos sollte man an jeden Servo anpassen. Dazu die Servowerte in die entsprechenden Regler eintragen. Momentan ist eingestellt 200 / 300 / 400, wobei 200 die rechte -, 300 die mittel- und 400 die linke Position ist.



- 1 Stromversorgung
- 2 TXT-Controller mit Programm
- 3 ft-Extender siehe Seite 8
- 4 Wandler 9V-5V
- 5 Servo PCA9685
- 6 Neige-Schwenk-Gehäuse mit 2 Servos und aufgeschraubten Raketenwerfer RC V959

Mini-I²C-Servo-Adapter

Für TX und TXT mit Levelshifter !



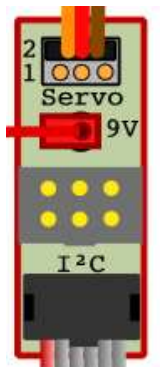
Der von Peter Habermehl entwickelte Servo-Adapter besitzt 2 Kanäle. Die Spannungsversorgung erfolgt über die 9 V Fischertechnikbuchse. Der I²C-Bus kann durchgeschleift werden.

Die Adresse ist 0x11 und die Kanäle sind 0x00 und 0x01.

Linksanschlag = 63

Mitte = 94

Rechtsanschlag = 125



Servo 2

Servo 1

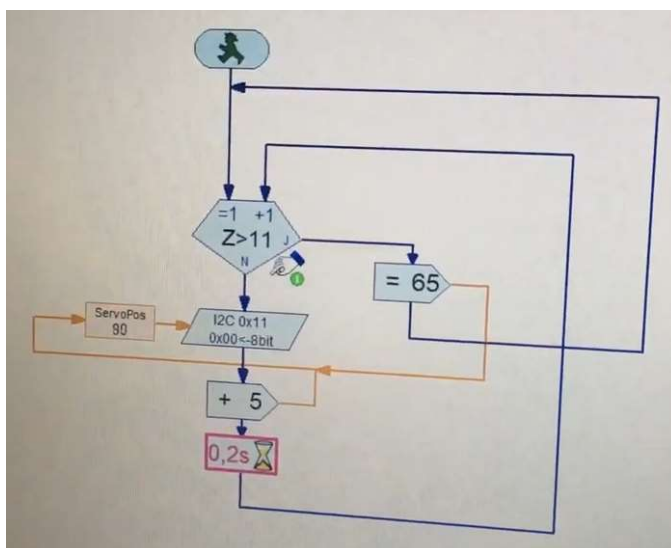
9 V Stromversorgung

I²C- Anschluss 1

I²C- Anschluss 2

I2C Schreiben	
Geräteadresse	0x11
Unteradresse	0x1
Dateneingang	<input checked="" type="checkbox"/>

I2C Schreiben	
Geräteadresse	0x11
Unteradresse	0x0
Dateneingang	<input checked="" type="checkbox"/>



Beispielprogramm für schrittweise Bewegung

OLED 128X64 Display SSD 1306



Für das kostengünstige OLED-Display gibt es von Cristian Bergschneider und Stefan Fuss unter <https://github.com/elektrofuzzis/oled> ein Beispielprogramm. Dort sind jede Menge Unterprogramme, von denen nur die relevanten besprochen werden

X1-0, Y1-0

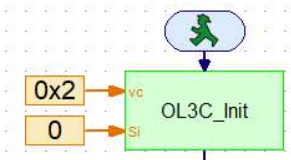


X2-127, Y2-63

Darstellbar sind:

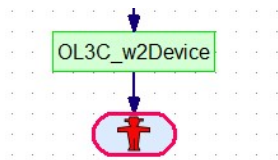
- Einzelne Pixel
- Zeichen in 8 Zeilen (0 bis 7) mit je 16 Zeichen
- Grafische Elemente innerhalb der angegebenen Werte für X und Y

Grundaufbau



Initialisierung des Displays. Einschalten und Bildschirm löschen

Hier können nun die gewünschten Elemente eingefügt werden. (siehe Tabelle)



Schreibt den Display Buffer ins OLED Display

Pixel Eingabe der Pixelposition	Linie Eingabe x1 u. y1 Anfang Eingabe x2 u. y2 Ende	Rechteck Eingabe x1 u. y1 li. oben Eingabe x2 u. y2 re. unten	Buchstabe Eingabe der Position x, y ch -Eingabe des Zeichens als Dez-Wert aus ASCII
Horizontale Linie x1,y1 – Startpunkt dx – Länge der Linie	Vertikale Linie x1,y1 – Startpunkt dy – Länge der Linie	Scrollen einschalten Di – Scrollrichtung St – siehe Anhang	Scrollen ausschalten Schaltet den Scrollmodus aus
Scroll nach re. oben 0x29 – Konstante	Scroll nach li. oben 0x2a – Konstante	Scroll nach rechts 0x26 – Konstante	Scroll nach links 0x27 – Konstante
Display Invers 0 – Normalansicht 1 – Inversansicht	Display löschen löscht den Display Buffer	Display an Schaltet das Display an	Display aus Schaltet das Display aus

Horizontal – jeweilige Zeile von - bis (St 0x0 bis St 0x7)

Vertikal – z.B. St 0x0 bis St 0xF nur nach oben,