

# Bedienungsanleitung Fridolin V 1.1



me.Jürgen Edel Hochschule Kaiserslautern Campus Pirmasens Carl-Schurz-Straße 10-16 66953 Pirmasens

email: juergen.edel@hs-kl.de

Idee und Umsetzung Jürgen Edel – Autor Jürgen Edel – Version 3.4

# Inhaltsverzeichnis

Abbildungsverzeichnis
Tabellenverzeichnis
Ansicht Fridolin
Belegung Dip-Schalter
1. Begriffserklärung
1.1 Mikrocontroller:
1.2 Prozessor:
1.3 I2C Bus:
1.4 Puls Weitenmodulation:
1.5 Flash- Speicher:
1.6 SRAM:
1.7 EEROM
2. Was ist ein Arduino
3. Arduino-Software installieren
4. Treiberinstallation
5. Com Port überprüfen 12
6. Arduino Entwicklungsumgebung13
7. Bibliothek einbinden
8. Arduino Nano V3.0 15
8.1 Arduino Nano Komponenten15
8.2 Arduino Nano V3.0 - PIN-Belegung16
8.3 Technische Daten Fridolin17
9. Programmcode
9.1 Taster Programmieren
9.2 Ultraschallsensor Programmieren19
9.3 Abstandssensoren Programmieren
9.4 Bodensensoren Programmieren21
9.5 Motoren Programmieren
9.6 LCD Anzeige Programmieren
9.7 Servo Programmieren
9.8 Akkuspannung berechnen 25
9.9 Akkuspannung über LCD Anzeigen 25
10. Akku Laden
11. Platine Bestückungsseite

12. Anhang	28
12.1 Tabelle TWI/I2C-Bus	29
12.2 Datenblatt Ultraschallmodul	30
12.3 Datenblatt Motortreiber	33
12.4 Datenblatt Servomotor	35
12.5 Datenblatt Sharp Abstandssensor	37
12.6 Datenblatt Bodensensor	46
12.8 Berechnung der TWI/I2C Adresse	47
12.7 Anschluss LCD Display	47

# Abbildungsverzeichnis

Abbildung 1: Arduino Nano	6
Abbildung 2: Arduino IDE	6
Abbildung 3: Installation starten	7
Abbildung 4: Istallation - Komponenten auswählen	7
Abbildung 5: Installation - Verzeichnis auswählen	8
Abbildung 6: Installation - Software installieren	8
Abbildung 7: Installation - Gerätesoftware	9
Abbildung 8: Installation bestätigen	9
Abbildung 9: Arduino Symbol	9
Abbildung 10: USB - Treiberinstallation	. 10
Abbildung 11: USB - Treiber Extrahieren	. 10
Abbildung 12: USB – Treiber	. 11
Abbildung 13: USB Treiber Setup	. 11
Abbildung 14: Com Port - Geräte und Drucker	. 12
Abbildung 15: Com Port - CH340	. 12
Abbildung 16: Arduino IDE - Erklärung	. 13
Abbildung 17: Arduino IDE - Board auswählen	13
Abbildung 18:ZIP - Datei	. 14
Abbildung 19:Bibliothek einbinden	. 14
Abbildung 20: Arduino Nano Komponenten	. 15
Abbildung 21: Arduino Nano Pin Belegung	. 16
Abbildung 22: Spannungsteiler	. 25
Abbildung 23: Akku Laden	. 26
Abbildung 24: Schaltbild Line Tracker	. 46
Abbildung 25: Schaltbild LCD / I2C Bus	. 47

# Tabellenverzeichnis

Tabelle 1: Motor- Logikpegel	16
Tabelle 2: TWI/I2C-Bus	29

# **Ansicht Fridolin**

- 1. Motorregler
- 2. Spannungsüberwachung Akku
- 3. Anschluss Signalleitungen
- 4. Anschluss Versorgungsspannung Sensoren
- 5. Dip Schalter
- 6. Arduino Nano
- 7. 5Volt Netzteil
- 8. LCD Display
- 9. Ladebuchse
- 10. Hauptschalter
- 11. Anzeige Akkuspannung
- 12. Ultraschall Sensor
- 13. IR Abstandssensor
- 14. Tastschalter
- 15. IR Bodensensoren
- 16. Akkupack
- 17. Servomotor
- 18. Antriebsmotoren



#### Belegung Dip-Schalter



#### Belegung Dip-Schalter

- 1. IR-Abstandssensor (mitte)
- 2. IR-Abstandssensor (links)
- 3. Bodensensor
- 4. Frei
- 5. IR-Abstandssensor (rechts)
- 6. Servomotor
- 7. Ultraschallsensor
- 8. LCD Display



# 1. Begriffserklärung

#### 1.1 Mikrocontroller:

Ein-Chip Computersystem bestehen aus einem Prozessor und anlogen sowie digitalen Ein und Ausgängen.

#### 1.2 Prozessor:

Programmierbares Rechenwerk Ein Prozessor kann über eigegebene Befehle Maschinen und elektronische Schaltungen steuern.

#### 1.3 I2C Bus:

I2C Bus<sup>2</sup>C ist ein Master-Slave-Zeidraht Bus.
Die Bezeichnung steht für Inter-Integrated Circuit.
Es gibt mindestens ein Master und mehrere Slaves.
Die Slaves können über eine eindeutige Adresse angesprochen werden.
TWI (two wire interface) ist ein vergleichbares System.

#### 1.4 Puls Weitenmodulation:

Die Pulsweitenmodulation wird zur Ansteuerung der Motoren verwendet. Ein Rechtecksignal mit einer Festen Frequenz variiert in der Impulsdauer.



Motor ist kurz eingeschaltet

Motor ist lange eingeschaltet

Beim Arduino liegt der PWM Wert zwischen 0 und 255

0 = keine Leistung

255= volle Leistung

#### 1.5 Flash- Speicher:

Nichtflüchtiger digitaler Speicherbaustein für größere Datenmengen Informationen bleiben nach abschalten erhalten

#### 1.6 SRAM:

Flüchtiger digitaler Speicherbaustein Informationen gehen nach abschalten verloren

#### 1.7 EEROM

Nichtflüchtiger digitaler Speicherbaustein für kleine Datenmengen Informationen bleiben nach abschalten erhalten

### 2. Was ist ein Arduino

Ein Arduino ist ein aus Hard und Software bestehendes interaktives, physisches System.

Die Software ist eine Open Source (" offene Quelle") Software deren

Quelltext öffentlich von Dritten benutzt werden darf.

Die Hardware in unserem Fall der "Arduino Nano" (Bild1) ist ein Ein/Ausgabeboard mit einem Mikrocontroller (MCU).

Die Arduino eigene integrierte Entwicklungsumgebung (IDE) (Bild2) ist eine Java Anwendung die für Windows, Mac und Linux kostenlos zur Verfügung steht.

Der Arduino kann in C, C++ und Assembler programmiert werden.



Abbildung 1: Arduino Nano

∞	
Datei Bearbeiten Sketch Werkzeuge Hilfe	
Blink§	
/* LED auf dem Board blinkt im Sekundentakt */	
<pre>// Anschlusspin deklarieren void secup() {     // Pin 13 Ausgang     pinMode(13, OUTPUT); }</pre>	
<pre>void loop() {     digitalWrite(13, HIGH); // LED einschalten     delay(1000); // 1 sek.warten     digitalWrite(13, LOW); // LED ausschalten     delay(1000); // 1 sek warten }</pre>	

Abbildung 2: Arduino IDE

# 3. Arduino-Software installieren

Laden Sie sich das Windows Installer Packet auf (https://www.arduino.cc/en/main/software) herunter und starten Sie die Installation.

Ist der Download abgeschlossen starten Sie die Installationsdatei



Folgen Sie der Installationsanweisung

💿 Arduino Setup: Installation Options	- 🗆 X	
Check the components you want to you don't want to install. Click Next	tall and uncheck the components continue.	
Select components to install: Ins Ins Cre Ass	Arduino software USB driver 2 Start Menu shortcut 2 Desktop shortcut ate .ino files	
Space required: 392.7MB		
Cancel Nullsoft Install System v	6 < <u>B</u> ack <u>N</u> ext >	

Abbildung 4: Istallation - Komponenten auswählen

Wählen Sie die zu installierenden Komponenten aus

💿 Arduino Setup: Installation Folder	_		×
Setup will install Arduino in the following folder. T folder, dick Browse and select another folder. C installation.	To install i lick Install	n a differen I to start the	e e
Destination Folder			
C:\Program Files (x86)\Arduino\		Browse	
Space required: 392.7MB			
Space available: 24.6GB			
Cancel Nullsoft Install System v2.46	< <u>B</u> ack	Inst	

Abbildung 5: Installation - Verzeichnis auswählen

Installationsverzeichnis auswählen (Standardeinstellungen beibehalten)

<u></u>	Arduino Setup: Installing -	- 🗆 🗙	
Size:6384	ł Kb Files: 15 Folders: 16		_
Show details	]		
Cancel	Nullsoft Install System v3,0 < Back	Close	

Abbildung 6: Installation - Software installieren

Software wird installiert

#### Erlauben Sie die Installation der Gerätesoftware



Abbildung 7: Installation - Gerätesoftware

<u></u>	Arduino Setup: Cor	mpleted <del>–</del>	
Completed			
Show details			
			t v
Cancel	Nullsoft Install System v3.0	< Back	Close

Abbildung 8: Installation bestätigen

#### Bestätigen sie die Installation.

Die Installation ist beendet und auf dem Desktop befindet sich ein Arduino Symbol



Jürgen Edel Hochschule Kaiserslautern

### 4. Treiberinstallation

Um mit dem Roboter über den USB Anschluss zu kommunizieren, müssen Sie noch einen zusätzlichen USB Treiber "CH341SER" installieren. Den Treiber finden Sie auf dem BSCW Server unter......

Entpacken Sie den ZIP komprimierten Ordner

Schriftan	t		Öffnen		
· · · 2 ·	1.1		In neuem Fenster öffnen		
			Alle extrahieren		
			An "Start" anheften		
	Tool	٣	Scannen auf Bedrohungen		
nsicht			Öffnen mit		
			Freigeben für		
ainfügen	Ver		Senden an	▶ leu	
ennugen			Ausschneiden	rar	
			Kopieren	-	
PC PD	own		Verknüpfung erstellen	-	
			Löschen		
Name			Umbenennen	eru	I
🔒 СНЗ	41SF		Eigenschaften	12	Abbildung 10: USB - Treiberinstallatio

n	×	
	😑 🖪 ZIP-komprimierte Ordner extrahieren	
		1
n	Wählen Sie ein Ziel aus und klicken Sie auf "Extrahieren".	
t	Dateien werden in diesen Ordner extrahiert:	
I	C:\Users\Jürgen Edel\Downloads\CH341SER Durchsuchen	I
ł		
I	✓ Dateien nach Extrahierung anzeigen	
1		
ł		
I		
I		
1		I
/	_	
1		I
4		
,		
1	Extrahieren Abbrechen	

Bild 1.9 USB – Treiber Extrahieren

Abbildung 11: USB - Treiber Extrahieren

#### Öffnen Sie den Ordner CH341SER



Abbildung 12: USB – Treiber

Installieren Sie den Treiber durch starten der Setup Funktion

1	📮 📜 👻	Tools für komprimierte Ordner	Downloads	_ 🗆 🗡
	📭   💽 🖪 = I	- 🗆 🗙		
	Datei Start Freigeben	Ansicht		^ <b>?</b>
Ko	Kopieren Einfügen Zwischenablage	iden eren ung einfügen Verschieben Kopieren nach * nach * Organisieren	Imbenennen Neuer Ordner Neu Öffnen	<ul> <li>Alles auswählen</li> <li>Nichts auswählen</li> <li>Auswahl umkehren</li> <li>Auswählen</li> </ul>
	🗲 🌛 🗸 🕇 📕 🕨 Die	ser PC → Downloads → CH341SER → CH	H341SER ▶	R" durchsuchen
	🗖 🖌 🐚			📋 🗙 🗸 🖃 🕥
	^	Name	Änderungsdatum Typ	Größe
∠ Typ	Dieser PC Bilder Dokumente Downloads Musik Videos Win 8.1 (C)		16.01.2017         14:44         Anwendungserw           16.01.2017         14:44         Systemdatei           16.01.2017         14:44         Systemdatei           16.01.2017         14:44         Sicherheitskatalo           16.01.2017         14:44         Setup-informatii           16.01.2017         14:44         Setup-informatii           16.01.2017         14:44         Systemdatei           16.01.2017         14:44         Systemdatei           16.01.2017         14:44         Systemdatei           16.01.2017         14:44         Virtueller Gerätet           16.01.2017         14:44         Anwendung	et 7 KB 57 KB 20 KB g 10 KB on 6 KB 39 KB re 20 KB 82 KB
	9 Elemente (Freier Speicherpla	tz: 478 GB)	238 KB 👘 Cor	mputer

Abbildung 13: USB Treiber Setup

#### Bestätigen Sie die Installation



# 5. Com Port überprüfen

Verbinden Sie den PC mittels des USB Kabel mit dem Roboter und schalten ihn ein. Über die Funktion ,

"Start- Geräte und Drucker" können Sie nun den com Port überprüfen



Abbildung 14: Com Port - Geräte und Drucker

Geräte und Drucker	- • ×
🔄 🎯 👻 🛧 🕷 « Alle Systemsteuerungselemente 🕨 Geräte und Drucker	✓ ♂ "Geräte und Drucker" durchsuc
Datei Bearbeiten Ansicht Extras ?	🔲 👗 🖻 📋 🗡 🗸 🖃 🌰
Gerät hinzufügen Drucker hinzufügen	I • 0
<ul> <li>PSATI-NB-ASUS USB Optical Mouse</li> <li>Nicht angegeben (1)</li> <li>UCB CEIMA</li> </ul>	^
CH340 (COM4)	
7 Elemente	
7 Elemente	r Computer

Abbildung 15: Com Port - CH340

# 6. Arduino Entwicklungsumgebung



#### Abbildung 16: Arduino IDE - Erklärung

Board auswählen Werkzeuge-rechte Maustaste Board: Arduino Nano

Diltraschall_08_09_16   Arduino			
Datei Bearbeiten Sketch Werkzeuge Hilfe			
🗸 📀 🗈 🔛 🛛 Automatische Formatierung	Strg+T		
Ultraschall_08_09_16 Sketch archivieren			
Kodierung korrigieren & neu lac	len		
Messgenauigkeit ca.lcm Serieller Monitor	Strg+Umschalt+M		
* Bei Messung über 1Meter * US trigger an Pin D7 Serieller Plotter	Strg+Umschalt+L		
* US echo an Pin D8 * Jürgen Edel HS Pirmagen	•	•	Boardverwalter
* vers.08_09_16 Prozessor: "ATmega328"	•		Arduino AVR-Boards
* Trigger an Nano D7 Port	•		Arduino Yún
* Echo an Nano D8 * Ausgabe über Seriellen h			Arduino/Genuino Uno
*/ Programmer: AVR ISP	•		Arduino Duemilanove or Diecimila
/ Pin zuweisen BOOtloader Drennen onst int trigger=7; // trigger an pin D7		•	Arduino Nano
onst int echo=8; // echo an pin D8			Arduino/Genuino Mega or Mega 2560
/ Variable zuweisen			Arduino Mega ADK
nt entfernung=0; // entfernung in cm			Arduino Leonardo
/ Ein und Ausgänge deklarieren			Arduino/Genuino Micro
void setup() { Serial.begin (9600); // Serielle datenübertragung starten pisModertinger. OUTPUT1; // Pin D2 als augagang delarieren			Arduino Esplora
			Arduino Mini
<pre>pinMode(echo, INPUT); // Pin D8 als Eingang deklarieren</pre>			Arduino Ethernet
/			Arduino Fio
oid loop()			Arduino BT
<pre>digitalWrite(trigger, LOW); // Trigger auf Low setzen</pre>			LilyPad Arduino USB
<pre>delayMicroseconds(2); digitalWrite(trigger, HIGH); // für 5 Microsekunden signal senden</pre>			LilyPad Arduino
<pre>delayMicroseconds(5); digitalWrite(trigger, LOW); // trigger abschalten</pre>			Arduino Pro or Pro Mini
<pre>dauer = pulseIn(echo, HIGH); // dauer des Echosignals messen /* Die Schallgeschuidinkeit liegt bei 340m/s ist cs. 29 Milwooshu</pre>	den nro cm		Arduino NG or older
<ul> <li>Der Impuls läuft hin und zurück, da aber nur die einfach Streck</li> </ul>	ke benötigt wird		Arduino Robot Control
* naide Strecke / 29 Microsekunden */			Arduino Robot Motor
			Arduino Gemma

Abbildung 17: Arduino IDE - Board auswählen

### 7. Bibliothek einbinden

Da die meisten Bibliotheken als Zip Datei vorliegen, kann man folgendermaßen vorgehen.

<u>Über den Menüpunkt</u> Sketch Bibliothek einbinden Zip-Bibliothek hinzufügen

Die entsprechende Zip- Datei suchen und über den Button öffnen bestätigen



Abbildung 19:Bibliothek einbinden



Abbildung 18:ZIP - Datei

# 8. Arduino Nano V3.0

# 8.1 Arduino Nano Komponenten Anschluss ISP Programmiergerät Power LED RX, TX LED Leuchten bei Serieller Datenübertragung verbunden Reset Taste Microcontroller MCU

LED Intern mit Pin D13

**USB** Anschluss

Abbildung 20: Arduino Nano Komponenten

#### 8.2 Arduino Nano V3.0 - PIN-Belegung



Abbildung 21: Arduino Nano Pin Belegung

Taster 1-3 , Low aktiv Motor A, rechts Motor B, links PWM Signal Motorsteuerung (255= volle Ausgangsspannung) (127= halbe Ausgangsspannung) (0 = Motor aus keine Spannung)

#### Logikpegel Motorsteuerung

	1 0		0
IA	IB		
L	L	Motor aus	
Н	L	Vorwärts	
L	Н	Rückwärts	
Н	Н	Motor aus	

Tabelle 1: Motor- Logikpegel

http://2.bp.blogspot.com/-\_M5HsO7r1dM/UGqw8xi2zUI/AAAAAAAR5c/O-Q5qzO1mJQ/s1600/arduino+pinout.png

Jürgen Edel Hochschule Kaiserslautern

#### 8.3 Technische Daten Fridolin

1	Mikrocontroller	ATmege328
	Betriebsspannung	5 Volt
1	Flash Speicher	32KB
1	SRAM	2КВ
1	EEROM	1KB
	Taktfrequenz	16 MHz
8	Analoge I/O Pins	8 Ein/Ausgänge,
13	Digitale I/O Pins	13 davon 6 PWM fähig
	Stromaufnahme	19mA
	Digitale Ein/Ausgänge	D0-D13
6	PWM fähig	D3, D5, D6, D9, D10, D11
8	Analoge Eingänge	A0-A7
3	Abstandssensoren	Infrarot Sensoren-Sharp 2Y0A41- Reichweite 3
		bis 30cm
1	Abstandssensoren	Ultraschallmodul HC-SR04 Reichweite 2 bis
		300 cm
3	Bodensensoren	Infrarot Sensoren-Fundoino Lines Tracker
		Modul TCRT 5000
1	Servomotor	Micro- Servo SG90
2	Antriebsmotoren	Gleichspannung 3 bis 12 Volt // Laststrom max
		300mA
1	LCD Display	Display Modul 1602 HD44780 // Zwei Zeilen
		16 Zeichen
1	I2C / TWI Controller	I2C Bus Modul PCF8574
1	Motortreiber	Treiber Modul L9110 // 3 bis 12 Volt 800mA
1	Netzteil	Step Down Modul LM2596
8	Akku 9,6 Volt	8 Mignonzellen 1,2 Volt 2700mA

Die PWM Signale werden durch einen 8bit Timer erzeugt.

Die analogen sowie digitale Ein/Ausgänge können mit 40mA belastet werden.

Bei den Ein/Ausgängen handelt es sich um 10bit AD-Wandler

An PIN D13 des Arduino Boards ist eine LED angeschlossen, die zu Testwecken programmiert werden kann.

# 9. Programmcode

```
9.1 Taster Programmieren
```

```
/*
Jürgen Edel
HS Pirmasens
Das drücken der Taster lässt die LED auf dem Arduino Board unterschiedlich blinken
Taster T1 an D2
Taster T2 an D12
Taster T3 an D4
Taster 1 LED Blink schnell
Taster 2 LED Blinkt langsam
Taster 3 LED Leuchtet
*/
// PIN zuweisen
const int Taster2=2; // Eingangspin für Taster Nr.2 ist D2
const int Taster1=12; // Eingngspin für Taster Nr.1 ist D12
const int Taster3=4; // Eingangspin für Taster Nr.3 ist D4
const int LED=13; // Ausgangspin für LED ist D13
// Ein und Ausgänge deklarieren
void setup()
{
 pinMode(Taster1,INPUT); // Taster Nr.1 als Eingang deklarieren
 pinMode(Taster2,INPUT); // Taster Nr.2 als Eingang deklarieren
 pinMode(Taster3,INPUT); // Taster Nr.3 als Eingang deklarieren
 pinMode(LED,OUTPUT); // Pin D13 als Ausgang definiert
}
     -----Hauptprogramm------
// ----
void loop()
{
 if(digitalRead(Taster1)==LOW) // Solange Taster 1 gedrückt ist führe Schleife aus
 {
  digitalWrite(LED,HIGH);
                               // LED Einschalten
                               // Pause 100 Millisekunden
  delay(100);
  digitalWrite(LED,LOW);
                              // LED Ausschalten
  delay(100);
                              // Pause 100 Millisekunden
 }
 if(digitalRead(Taster2)==LOW) // Solange Taster 2 gedrückt ist führe Schleife aus
                              // LED Einschalten
 digitalWrite(LED,HIGH);
 delay(1000);
                              // Pause 1 Sekunde
 digitalWrite(LED,LOW);
                              // LED Ausschalten
                              // Pause 1 Sekunde
 delay(1000);
 }
 if(digitalRead(Taster3)==LOW) // Solange Taster 3 gedrückt ist führe Schleife aus
 {
 digitalWrite(LED,HIGH);
                              // LED Einschalten
 digitalWrite(LED,LOW);
                              // LED Ausschalten
}
```

#### 9.2 Ultraschallsensor Programmieren

```
/* Ultraschall Testprogramm
* Jürgen Edel hs-kl
* trigger an D7
* echo an D8
 */
// Pin zuweisen
const int trigger=7;
                         // trigger an pin D7
const int echo=8;
                         // echo an pin D8
// Variable zuweisen
int dauer=0;
                         // Zeitdauer des US impulses
// Ein und Ausgänge deklarieren
void setup()
{
 pinMode(trigger, OUTPUT); // Pin D7 als Ausgang delarieren
 pinMode(echo, INPUT);
                              // Pin D8 als Eingang deklarieren
}
// Hautprogramm
void loop()
{
 digitalWrite(trigger, LOW);
                              // Trigger auf Low setzen
 delayMicroseconds(2);
 digitalWrite(trigger, HIGH);
                               // für 5 Microsekunden signal senden
 delayMicroseconds(5);
 digitalWrite(trigger, LOW);
                              // trigger abschalten
 dauer = pulseIn(echo, HIGH); // dauer des Echosignals messen
                              // Pause von einer sekunde bis zur nächsten messung
 delay(1000);
```

```
9.3 Abstandssensoren Programmieren
```

/\* Entfernungsmessung mit Sharp IR Sensor 2Y0A41

- \* Spannung über analogPort einlesen
- \* Sensor links an A2
- \* Sensor rechts an A0
- \* Sensor mitte an A1
- \* Jürgen Edel HS-KL

Serial.begin (9600);

```
*/
```

```
// Anschlusspin zuweisen
const int SL=A2;
                   // Anschluss linker Sensor an Pin A2
const int SR=A0;
                   // Anschluss rechter Sensor an Pin A0
                  // Mittlerer Sensor Pin A1
const int SM=A1;
// Variable zuweisen
                 // Wert Sensor links
int WSL=0;
int WSR=0;
                 // Wert Sensor rechts
int WSM=0;
                 // Wert Sensor mitte
void setup()
{
```

WSL=analogRead(SL); // spannung linker sensor einlesen WSR=analogRead(SR); // spannung rechter sensor einlesen WSM=analogRead(SM); // spannung mittlerer sensor einlesen

// 1 sek. Pause bis zur nächten Messung

}

{

void loop()

delay(1000);

#### 9.4 Bodensensoren Programmieren

```
/*
```

- \* Testprogramm Bodensensoren
- \* Sensor rechts an D11
- \* Sensor links an D13
- \* Sensor mitte an A3
- \* Sensoren rechts und links erkennen nur 1(High) oder 0(LOW)
- \* Schwarzer untergrung ist O(LOW)
- \* Sensor mitte liefert einen Zahlenwert von 0-1023
- \* Schwarzer Untergrung hoher Zahlenwert, heller Untergrund kleiner Zahlenwert
- \* HS-KL / Jürgen Edel

```
*/
```

// PIN zuweisung Bodensensor

const int BSR=11;	// Bodensensor rechts D11
const int BSM=A3;	// Bodensensor mitte A3
const int BSL=13;	// Bodensensor links D13

// Variable zuweisen

int BL;	<pre>// Variable Bodensensor links</pre>
int BM;	// Variable Bodensensor mitte
int BR;	// Variable Bodensensor rechts

```
// Ein und Ausgänge deklarieren
void setup()
{
   Serial.begin (9600);
   pinMode(BSR,INPUT); // Bodensensor rechts Pin D11 Eingang
   pinMode(BSM,INPUT); // Bodensensor mitte Pin A3 Eingang
   pinMode(BSL,INPUT); // Bodensensor links Pin D13 Eingang
```

}

```
//-----Hauptprogramm-----
void loop()
{
    BR=digitalRead(BSR); // Einlesen der Information vom rechten Bodensensor
    BM=analogRead(BSM); // Einlesen der Information vom mittleren Bodensensor
    BL=digitalRead(BSL); // Einlesen der Information vom linken Bodensensor
    delay(500); // Eine halbe Sekunde Pause bis zur nächten Messung
```

#### 9.5 Motoren Programmieren

/\* Jürgen Edel **HS** Pirmasens Testprogramm geradeausfahrt \*/ // A=rechter Motor const int AIA=9; // Zuweisung Digital Pin D9 PWM const int AIB=6; // Zuweisung Digital Pin D6 PWM // B=linker Motor const int BIA=5; // Zuweisung Digital Pin D5 PWM const int BIB=3; // Zuweisung Digital Pin D3 PWM /\* Variable zuweisen \* PWM Impuls von 255 entspricht volle Ausgangsspannung des Motortreibers \* ist 100% Motorleistung \*/ int speedR=255; // Geschwindigkeit Motor Rechts int speedL=255; // Geschwindigkeit Motor Links // Ein und Ausgänge deklarieren void setup() { pinMode(AIA,OUTPUT); // PWM pinMode(AIB,OUTPUT); // PWM pinMode(BIA,OUTPUT); // PWM pinMode(BIB,OUTPUT); // PWM } // ------Hauptprogramm-----void loop() { // vorwärts fahren analogWrite(AIA, speedR); // Geschwindigkeit für rechten Motor analogWrite(AIB,0); // Drehrichtung vorwärts analogWrite(BIA,speedL); // Geschwindigkeit für linken Motor analogWrite(BIB,0); // Drehrichtung vorwärts

#### 9.6 LCD Anzeige Programmieren

Die LCD Anzeige wird über einen I2C-Bus angesteuert. Um die Anzeige bzw. den Bus anzusprechen erfolgt die Programmierung über eine eindeutige Adresse die am Slave (I2C-Bus) eingestellt ist. <u>Siehe Tabelle TWI/I2C Bus</u> <u>Berechnung Adresse</u>

Achtung

/\* LCD Test TWI Controller Jürgen Edel HS-KL \*/

// Bibliothek einbinden
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal\_I2C.h>

In den Bibliotheken Ordner müssen noch zwei zusätzliche Bibliotheken aufgenommen werden. LiquidCrystal I2C NewliquidCrystal 1-3-4 Siehe Seite 14

/\*

Definieren der Pins am TWI bzw. I2C Bus-Controller Adresse des Displays mit PCF8574 AT. Bei PCF8574A muss die Adresse auf 0x27 geändert werden \*/ define I2C\_ADDR 0x3F // Adresse I2C Bus (TWI) #define Hintergrund 3 #define En pin 2 #define Rw pin 1 #define Rs\_pin 0 #define D4\_pin 4 #define D5 pin 5 #define D6\_pin 6 #define D7\_pin 7 LiquidCrystal\_I2C lcd(I2C\_ADDR,En\_pin,Rw\_pin,Rs\_pin,D4\_pin,D5\_pin,D6\_pin,D7\_pin); void setup() { lcd.begin (16,2); // 16 Zeichen, 2 Zeilen } // Hauptprogramm void loop() { // Beleuchtung Anschalten lcd.setBacklightPin(Hintergrund, POSITIVE); lcd.setBacklight(HIGH); // Ausgabe auf Dislpay lcd.setCursor (0,0); // Cursor auf erste Zeile lcd.print("8888888888888888"); // Text ausgeben lcd.setCursor (0,1); // Cursor auf zweite Zeile lcd.print("888888888888888888"); // Text ausgeben delay(1000); } Jürgen Edel

#### 9.7 Servo Programmieren

```
/*
```

Servotest 0-180 grad von rechts nach links und zurück HS-KL / Jürgen Edel

#### \*/

```
// Pinbelegung festlegen
const int servo=10; // Anschlusspin für Servomotor ist Pin D10
// Variable Servo
int servowert;
                // Servowert in Microsekunden
void setup()
{
pinMode(servo,OUTPUT); // Pin D10 als Ausgang deklarieren
}
// ------Hauptprogramm------
void loop()
{
for(int winkel=180; winkel>0; winkel--) // Zählschleife
 {
 servowert=winkel;
                                           // Ausgangspin für Servo D10 auf High setzen
 digitalWrite(servo,HIGH);
 servowert=map(servowert,0,180,800,2500);
                                           // Winkel in Zeit Umrechnen
 delayMicroseconds(servowert);
                                           // Pause berechnete Zeit
 digitalWrite(servo,LOW);
                                           // Ausgangspin für Servo D10 auf Low setzen
 delay(20);
                                           // 20 millisekunden Pause
 }
```

}// ende Loop

#### 9.8 Akkuspannung berechnen



Der Controller verfügt über einen 10bit analog Digital Wandler mit dem die Spannung am 47k Ohm Widerstand gemessen wird.

Folglich muss der gemessene Analogwert in einen Spannungswert umgerechnet werden.

#### 9.9 Akkuspannung über LCD Anzeigen

/\*

- \* HS-KL / Jürgen Edel
- \* Akkuspannung Messen
- \* Akku (Spannungsteiler an A6)
- \* Spannungsteiler 100k zu 47k
- \* analogwert wird an 47k Widerstand gemessen
- т ... /
- \*/

// Anschlusspin Spannungsteiler zuweisen "Spannungsteiler 100k zu 47k"
const int Batterie=A6; // Anschlusspin für Spannungsteiler Pin6 (A6)

// Variable zuweisen	
float Messen=0;	// Variable Wert am Spannungsteiler (Spannung über 47 kOhm )
float Batt=0;	// Umrechnung des Analogwertes in Volt

void setup()

```
{
```

pinMode(Batterie,INPUT); // Eingangspin für Spannungsteiler A6 als Eingang definieren }

```
// -----Hauptprogramm------
void loop()
{
```

```
Messen=analogRead(Batterie); // Messen der Spannung am Spanungsteiler
Batt=(Messen*147)/47*5/1023; // Berechnung der Batteriespannung in Volt
}
```

# 10. Akku Laden

# **Fridolin Laden**

# Ladevorgang

Der Akku-Pack wird solange geladen, solange der Akku am Ladegerät angesteckt ist. Um eine Beschädigung und Überladung des Akku-Packs zu vermeiden, ist der Ladevorgang nach <u>maximal 24 Std.</u> zu beenden.

vernielden, ist der Eddevorgung nach <u>maximar 24 star</u> zu seender

#### Das Fahrzeug bitte erst Laden wenn die Kontroll- LED leuchtet



Jürgen Edel Hochschule Kaiserslautern

# **11. Platine Bestückungsseite**



# 12. Anhang

#### Bezugsquellen

<u>Bücher:</u> Die elektronische Welt mit Arduino entdecken *Erik Bartmann* 

Arduino Kochbuch Michael Margolis

Arduino für Einsteiger Massimo Banzi

Internet: Arduino Forum https://www.arduinoforum.de/

Arduino Tutorial www.arduino-tutorial.de/

Arduino Spielwiese www.arduinospielwiese.de/

#### 12.1 Tabelle TWI/I2C-Bus

	TWI / I2C-Bus	TWI / I2C-Bus
	Adresse	Adresse
Fahrzeug Nr.	0x3F	0x27
1		Х
2		Х
3	Х	
4	Х	
5	Х	
6	Х	
7	Х	
8	Х	
9	Х	
10		Х
11	Х	
12		Х
13	Х	
14	Х	
15	Х	
16	Х	
17	Х	
18	Х	
19	Х	
20	х	

Tabelle 2: TWI/I2C-Bus

Zum Ansteuern des LCD Displays muss die entsprechende Bus-Adresse im Programmcode eingegeben werden.

Programmcode:

LCD Anzeige programmieren Seite 23

# Ultraschall Messmodul HC-SR04

#### Beschreibung:

Das Ultraschall Modul HC-SR04 eignet sich zur Entfernungsmessung im Bereich zwischen 2cm und ca. 3m mit einer Auflösung von 3mm. Es benötigt nur eine einfache Versorgungsspannung von 5V bei einer Stromaufnahme von <<u>2mA</u>. Nach Triggerung mit einer <u>fallenden Flanke</u> (TTL - Pegel) misst das Modul selbstständig die Entfernung und wandelt diese in ein <u>PWM Signal</u> welches am Ausgang zur Verfügung steht. Ein Messintervall hat eine Dauer von 20ms. Es können also 50 Messungen pro Sekunde durchgeführt werden.

Anwendungsbereiche: Hinderniserkennung, Entfernungsmessung, Füllstandanzeiger, Industrieanwendungen.

# **KT-elektronic**



#### Pinbeschreibung:

- 1: VCC, Versorgungsspannung 5V
- 2: Triggereingang, TTL-Pegel 3: Echo, Ausgang Messergebnis, TTL-Pegel
- 4: GND, 0V

Power Supply	VCC +5V +-10%, GND 0V
Signal Level (Trigger, Echo)	TTL Pegel (L < 1,0V)
messbare Distanz	2cm - ca. 300cm
Messintervall	0,3cm
Messungen pro Sekunde	maximal 50
Abmessungen (I, b, t) mm	45 x 21 x 18

#### Anwendung:

Das Auslösen eines Messzyklus geschieht durch eine fallende Flanke am Triggereingang (Pin 2) für mindestens 10µs. Das Modul sendet darauf nach ca. 250µs ein 40 kHz Burst-Signal für die Dauer von 200µs. Danach geht der Ausgang (Echo, Pin 3) sofort auf H-Pegel und das Modul wartet auf den Empfang des Echos. Wird dieses detektiert fällt der Ausgang auf L-Pegel. 20ms nach Triggerung kann eine weitere Messung stattfinden. Wird kein Echo detektiert verweilt der Ausgang für insgesamt 200ms auf H-Pegel und zeigt so die erfolglose Messung an. Danach wartet das Modul auf die nächste fallende Flanke am Triggereingang und die Messung beginnt neu. Die besten Messergebnisse ergeben sich bei Reflektion an glatten, ebenen Flächen. Bei Distanzen bis 1m ist das Material der Fläche recht unkritisch. Der Winkel zum Objekt kann bei kurzen Distanzen von unter 1m bis etwa 45° betragen. Auch recht dünne Objekte werden zuverlässig erkannt. Ein normaler Kugelschreiber z.B. lässt sich bis auf eine Distanz von ca. 30cm sicher erfassen. Bei der maximalen Distanz von 3m muss schon genau gezielt werden und es sollten keine anderen Gegenstände in ähnlicher Entfernung im Sendekegel von 15° vorhanden sein.

Januar 2012, Copyright © KT-Elektronic

tt

#### Ultraschall Messmodul HC-SR04

#### Single Shot Modus:

Im folgenden ist eine einzelne Messung zu sehen. Kanal 2 ist das Triggersignal, Kanal 1 der Ausgang. Die Laufzeit des Ultraschall-Burst beträgt bei dieser Messung etwa 600µs (200µs/Div). Die angezeigte Zeit von 456µs setzt sich aus dem 250µs Delay nach Triggerung und dem anschließenden 200µs Burst zusammen.



#### Auswertung der Messung:

Für die Berechnung der Entfernung zum Messobjekt wird nun die Schallgeschwindigkeit in Luft von 343m/S (bei 20°C) herangezogen. 343m/S entsprechen 34,3cm pro Millisekunde (ms)

Daraus ergibt sich für dieses Beispiel bei 600µS (0,6ms):

34,3cm x 0,6ms eine Laufstrecke von 20,6cm die das Signal zurückgelegt hat.

Da es sich um ein Echo handelt wird die Strecke zwei mal durchlaufen, weshalb wir den Wert halbieren müssen.

20,6cm / 2 = 10,3cm Abstand zum Messobjekt.

#### Kontinuierliche Messung:

Es besteht die Möglichkeit eine permanente Messung durchzuführen. Da das Modul auf die fallende Flanke des Triggereingangs reagiert ist es nicht nötig ein kurzes Triggersignal zu erzeugen. Es genügt ein Rechtecksignal mit einer Frequenz kleiner 50Hz (20mS Intervall) an den Triggereingang anzulegen um eine dauernde Messung durchzuführen.

Auf Kanal 2 liegt wieder der Triggereingang, das Rechtecksignal hat eine Frequenz von 25Hz die Amplitude beträgt 5V (Tastkopf 1:10).



#### Auswertung der Messung:

Bei dieser Messung beträgt das Intervall 40ms was 25 Messungen in der Sekunde entspricht. Das Messobjekt ist ca.180cm entfernt. Das Oszilloskop triggert diesmal auf die steigende Flanke von Kanal 1 und misst die Zeit bis zur Mittellinie. Die gemessene Laufzeit beträgt 10,4ms. Daraus ergibt sich eine Entfernung von:

10,4ms x 34,3cm / 2 = 178,3cm

Eine weitere Messung mit einer Zeitbasis von 1ms/Div um die Messgenauigkeit zu erhöhen (ohne Bild) ergab eine Zeit von 10,56ms was einer Entfernung von 181cm entspricht. Die tatsächliche Entfernung betrug bei dieser Messung 182,5cm was eine Genauigkeit von besser 1% ergibt.

tt

#### Ultraschall Messmodul HC-SR04

#### Messgenauigkeit:

Die systembedingte Messgenauigkeit beträgt ca. 3mm und hängt mit der internen Abtastrate des Moduls zusammen. Ein weiterer Faktor ist die Temperaturabhängigkeit der Schallgeschwindigkeit in Luft. Näherungsweise kann man die Schallgeschwindigkeit in Abhängigkeit der Temperatur im Bereich von -20°C bis +40°C mit folgender Formel berechnen:

$$c_{\text{Luft}} \approx (331,5+0,6 \,\vartheta/^{\circ}\text{C})\frac{m}{s}$$

Theta = Umgebungstemperatur in °C

Für 20°C Raumtemperatur ergibt sich also:  $c = 331,5 + (0,6 \times 20) = 343,5 \text{ m/s}.$ 

Die folgende Tabelle enthält einige Werte die rechnerisch für die Laufzeit zu erwarten sind:

Entfernung zum Objekt [cm]	Laufzeit [ms] bei 20°C	Laufzeit [ms] bei 0°C
2	0,117	0,121
10	0,583	0,603
50	2,915	3,017
100	5,831	6,033
200	11,662	12,066
300 (max. Dist.)	17,492	18,100

Es zeigt sich das bei 20°C Temperaturdifferenz ein Fehler von 3,4% entsteht. Bei der Verwendung im Freien und Messung von größeren Distanzen wäre also zu überlegen ob eine Temperaturkompensation sinnvoll ist.

tt

#### 12.3 Datenblatt Motortreiber

#### L9110 2-CHANNEL MOTOR DRIVER



#### Description

The L9110S 2-Channel motor driver module is a compact board that can be used to drive small robots. This module has two independent motor driver chips which can each drive up 800mA of continuous current. The boards can be operated from 2.5V to 12V enabling this module to be used with both 3.3V and 5V microcontrollers.

A set of female header pins is used to connect this module to a microcontroller. The motors are attached via two sets of screw terminals.

A PWM Pulse Width Modulation signal is used to control the speed of a motor and a digital output is used to change its direction. This module can also be used to drive a single four line two phase stepper motor. Four holes make this board easy to mount onto your robot or other project.

#### Specifications

- On-board 2 L9110 motor control chip
- Module can be driven by two dc motors at the same time or one phase 4 line 2 type stepping motor
- Input voltage: 2.5-12V DC
- Each channel has a continuous output current 800 ma
- PCB Size: 29.2mm x 23mm

#### **Pin Configuration**



- 1. B-IA: Motor B Input A
- 2. B-IB: Motor B Input B
- 3. GND: ground
- 4. VCC: 2.5V-12V DC
- 5. A-IA: Motor A Input A
- 6. A-IB: Motor B Input B

#### Logiktabelle Rechter Motor

AIA	AIB	
L	L	Motor aus
Н	L	Vorwärts
L	Н	Rückwärts
Н	Н	Motor aus

Logiktabelle Linker Motor

BIA	BIB	
L	L	Motor aus
Н	L	Vorwärts
L	Н	Rückwärts
Н	Н	Motor aus

# SG90 9 g Micro Servo



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

#### **Specifications**

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 µs
- Temperature range: 0 °C 55 °C



Position "0" (1.5 ms pulse) is middle, "90" ( $\sim$ 2 ms pulse) is all the way to the right, "-90" ( $\sim$ 1 ms pulse) is all the way to the left.

GP2Y0A41SK0F

# GP2Y0A41SK0F

Distance Measuring Sensor Unit Measuring distance : 4 to 30 cm Analog output type



#### Description

GP2Y0A41SK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IR-LED (infrared emitting diode) and signal processing circuit. The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

#### ■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

#### Applications

- 1. Cleaning robot
- 2. Personal robot
- 3. Sanitary

#### Features

- 1. Distance measuring sensor is united with PSD, infrared LED and signal processing circuit
- 2. Short measuring cycle (16.5ms)
- 3. Distance measuring range : 4 to 30 cm
- 4. Package size  $(29.5 \times 13.0 \times 13.5 \text{mm})$
- 5. Analog output type

Notice The content of data sheet is subject to change without prior notice. In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

1



■Schematic

#### 

■Outline



2



#### GP2Y0A41SK0F

#### ■Absolute maximum ratings

	linge			(Ta=25°C, Vcc=5V)
Parameter	Symbol	Ratings	Unit	Remark
Supply voltage	Vcc	-0.3 to +7	V	-
Output terminal voltage	Vo	-0.3 to Vcc+0.3	V	-
Operating temperature	Topr	-10 to +60	°C	
Storage temperature	Tstg	-40 to +70	°C	5

#### Operating supply voltage

Symbol	Rating	Unit	Remark	
Vcc	4.5 to 5.5	V	10 <b>0</b>	

#### ■Electro-optical Characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Measuring distance range	ΔL	(Note 1)	4		30	Cm
Output terminal voltage	Vo	L=30cm (Note 1)	0.25	0.4	0.55	V
Output voltage difference	ΔVo	Output change at L change $(30 \text{cm} \rightarrow 4 \text{cm})$ (Note 1)	1.95	2.25	2.55	V
Average supply current	Icc	L=30cm (Note 1)		12	22	mA

%L : Distance to reflective object (Note 1) Using reflective object : White paper

(Made by Kodak Co., Ltd. gray cards R-27 • white face, reflective ratio; 90%)

#### ■Timing Chart



3

#### ■Supplements

•Example of output distance characteristics



4



5

•Example of output distance characteristics with the inverse of distance

•This product shall not contain the following materials.

Also, the following materials shall not be used in the production process for this product.

- Materials for ODS : CFC<sub>S</sub>, Halon, Carbon tetrachloride 1.1.1-Trichloroethane (Methyl chloroform) •Product mass : Approx. 3.6g (TYP)
- •This product does not contain the chemical materials regulated by RoHS directive.
- (Except for the NOT regulated by RoHS directive.)

•Compliance with each regulation

1) The RoHS directive(2002/95/EC) This product complies with the RoHS directive(2002/95/EC).

Object substances: mercury, lead (except for lead in high melting temperature type solders<sup>\*1</sup> and glass of electronic components), cadmium, hexavalent chromium, polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE)

\*1<sup>+</sup> i.e. tin-lead solder alloys containing more than 85% lead

2) Content of six substances specified in Management Methods for Control of Pollution Caused by Electronic Information Products Regulation (Chinese: 电子信息产品污染控制管理办法).

	Toxic and hazardous substances					
Category	Lead (Pb)	Mercury (Hg)	Cadmium (Cd)	Hexavalent chromium (Cr <sup>6+</sup> )	Polybrominated biphenyls (PBB)	Polybrominated diphenyl ethers (PBDE)
Distance Measuring Sensor	*	1	1	1	1	1

✓: indicates that the content of the toxic and hazardous substance in all the homogeneous materials of the part is below the concentration limit requirement as described in SJ/T 11363-2006 standard.

\*: indicates that the content of the toxic and hazardous substance in at least one homogeneous material of the part exceeds the concentration limit requirement as described in SJ/T 11363-2006 standard. Lead in high melting temperature type solders (i.e. tin-lead solder alloys containing more than 85% lead) and glass of electronic components (designated by "\*" in the above table) are exempt from the RoHS directive (2002/95/EC), because there is no effective way to eliminate or substitute them by present scientific technology.

6

#### Notes

[Advice for the optics]

- •Lens of this device shall be kept cleanly. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- •In case that protection is set in front of the emitter and detector portion, the protection cover which has the most efficient transmittance at the emitting wavelength range of LED for this product ( $\lambda$ =870nm±70nm), shall be recommended to use. The face and back of protection cover should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied with according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.

[Advice for the characteristics]

- •In case that there is an object near to light exits of the sensor between the sensor and the detected object, please use this device after confirming sufficiently what the characteristics of this sensor do not change by the object.
- •When the detector surface receive direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector does not receive direct light from such light source.
- •Distance between sensor and mirror reflector can not sometimes measure exactly.
- In case of changing the mounting angle of this product, it may measure the distance exactly.
- •In case that reflective object has boundary line clearly, there is cases that distance can not measure exactly. At that time, if direction of boundary line and the line between emitter center and detector center parallels, it is possible to decrease deviation of measuring distance.
- it is possible to decrease de vidion of measuring distance



• In order to decrease measuring error by moving direction of object, we recommend to mount the sensor like below drawing.



•In order to stabilize power supply line, we recommend to connect a by-pass capacitor of  $10\mu$ F or more between Vcc and GND near this product.

[Notes on handling]

- •Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against for washing.
- •There are some possibilities that the sensor inside the case package with lens may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the case package with lens and also on the PCB at the assembly and inserting of the set.

7

#### Packing specification



8

#### Important Notices

 The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

 Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

 Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:

- --- Personal computers
- --- Office automation equipment
- --- Telecommunication equipment [terminal] --- Test and measurement equipment
- --- Industrial control
- --- Audio visual equipment
- --- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection with equipment that requires higher reliability such as: --- Transportation control and safety equipment (i.e.,

- aircraft, trains, automobiles, etc.)
- Traffic signals
- --- Gas leakage sensor breakers
- --- Alarm equipment
- --- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:

- --- Space applications
- --- Telecommunication equipment [trunk lines]
- --- Nuclear power control equipment
- --- Medical and other life support equipment (e.g., scuba).

 If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

 This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

#### 12.6 Datenblatt Bodensensor

Line Tracker





Abbildung 24: Schaltbild Line Tracker

An dem Bodensensor wurde an den Ausgang des mittleren Sensors eine zusätzliche Leitung angelötet, sodass je nach Untergrund und Reflektionsgrad ein analoger Wert abgenommen werden kann.

Da der Arduino Nano einen 10bit AD Wandler benutzt, liegt der Wert des Sensors zwischen 0 und 1023.

Die Kabelfarbe ist violett.

An dem rechten und linken Bodensensor kann über den Schmitt Trigger ein High oder ein Low Signal abgenommen werden.

Kabelfarbe Bodensensor links=orange

Kabelfarbe Bodensensor rechts=weis

#### 12.7 Anschluss LCD Display

#### Schaltbild LCD –I2C- Bus



Abbildung 25: Schaltbild LCD / I2C Bus

#### 12.8 Berechnung der TWI/I2C Adresse

#### Kommunikation - IC PCF8574

#### 7.1 Addressing

For addressing see Figs 9, 10 and 11.



Each of the PCF8574's eight I/Os can be independently used as an input or output. Input data is transferred from the port to the microcontroller by the READ mode (see Fig.11). Output data is transmitted to the port by the WRITE mode (see Fig.10).

Auszug aus Datenblatt PCF8574 Hersteller Philips

Zum Controller

### Umrechnung Binär in Dezimal und Hexadezimal



Binär	0100111	Binär	0111111	
Dezimal	39	Dezimal	63	
Hexadezimal	27	Hexadezimal	3F	
Adressierungsco	de	Adressierungscode		
"define I2C_ADD	R 0x27"	<pre>"define I2C_ADDR 0x3F"</pre>		

Je nach IC Typ wird das LCD Display mit einer anderen Adresse angesprochen

Siehe Seite 23 9.6 LCD Anzeige Programmieren