

1. August 2016

Roboter Abstandssensorik

Version 1

**HAW Hamburg
Deeply Embedded**

Edgar Toll

Änderungshistorie

Version	Author	Datum	Anmerkungen/ Änderungen
1	Edgar	01.08.16	initiale Version

Inhaltsverzeichnis

1	Einführung	1
2	Ansätze	2
2.1	Digitaler Sensor	2
2.1.1	Technische Daten	2
2.1.2	Pegelwandler	2
2.1.3	Anschluss an digitalen Pin	2
2.1.4	Ansteuerung per BeagleBone Black	2
2.1.5	Bewertung	3
2.2	SHARP GP2D12	5
2.2.1	Technische Daten	5
2.2.2	Anschluss an analogen Pin	5
2.2.3	Ansteuerung per BeagleBone Black	8
2.2.4	Bewertung	9
3	Vergleich der Ansätze	10
4	Zusammenfassung	11
5	Anhang	12
5.1	Elektrotechnische Grundannahmen	12
5.1.1	Parallelschaltung von Widerständen	12
5.2	Abbildungsverzeichnis	13
5.3	Tabellenverzeichnis	13
5.4	Listings	13
5.5	Quellen	14

1 Einführung

Ein kleiner, vierbeiniger, 3D gedruckter Roboter kann zwar laufen, weiß aber selbst noch nicht, ob sein Weg überhaupt frei ist.

Genutzt werden kann dafür Entfernungssensorik. In diesem Paper werden zwei Ansätze für Abstandssensorik verfolgt. Dabei wird, zusätzlich zur Funktionalität, auf Größe und Gewicht geachtet, da von kleinen Servos und wenig Platz ausgegangen wird.

Da die Servos mit einem BeagleBone Black angesteuert werden, wird die Sensorik die bestehende Plattform nutzen, um Gewicht und Platz zu sparen.

2 Ansätze

Für die Umsetzung der Hinderniserkennung wurden mehrere, auf infrarot basierende Ansätze verfolgt. In Unterabschnitt 2.1 wird ein simpler Sensor eines BeagleBone Sensor Sets getestet. Als professionelle Variante wird in Unterabschnitt 2.2 ein SHARP Sensor analysiert.

2.1 Digitaler Sensor

Um die erste Idee zu verwirklichen wurde ein digitaler Distanzsensor aus einem Sensor Set verwendet. Der Sensor zieht den Datenpin auf HIGH, solange sich etwas innerhalb von einer einstellbaren Entfernung befindet und umgekehrt auf LOW, wenn sich nichts in Reichweite befindet.

2.1.1 Technische Daten

- Digitaler Output
- Operationsspannung: 5V
- Entfernung: einstellbar zwischen 0cm bis 40cm

2.1.2 Pegelwandler

Da der Sensor auf 5V Basis arbeitet und das BeagleBone Black auf 3,3V Basis arbeitet, muss die Spannung herunter geregelt werden. Um dies zu realisieren wird ein Pegelwandler (engl. Logic Level Converter), wie in Abbildung 1 (S. 2) zu sehen, verwendet. Der Pegelwandler hat dabei eine High Voltage (HV) und eine Low Voltage (LV) Seite. Die jeweils höhere der beiden Spannungen wird an die HV Seite angelegt. Die zusammengehörigen Datenpins sind immer gegenüberliegend und können jeweils in beide Richtungen arbeiten. Verwendet wurde ein Pegelwandler von SparkFun [2].

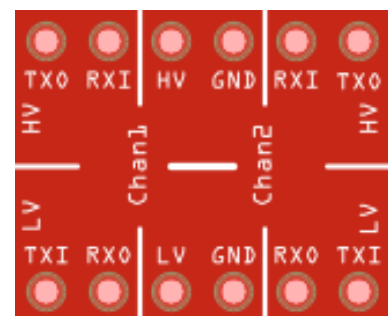


Abbildung 1: Pegelwandler [2, 5]

2.1.3 Anschluss an digitalen Pin

Die Beschaltung ist in Abbildung 2 (S. 3) zu sehen. Dabei werden die HV und LV Seite jeweils mit ihrer jeweiligen Spannung (dunkelrot 5V, orange 3,3V, schwarz GND) beschaltet. Einer der LV Datenpins wird an den digitalen Pin (P9_12) angeschlossen (gelb) und der, auf der HV Seite gegenüberliegende Datenpin wird an den Datenausgang des Sensors angeschlossen (gelb).

2.1.4 Ansteuerung per BeagleBone Black

Um die Machbarkeit zu zeigen wird im ersten Schritt nur mit Hilfe eines kleinen BoneScript Programms in Listing 1 (S. 3) der Datenpin (P9_12) abgefragt. In einem späteren Schritt

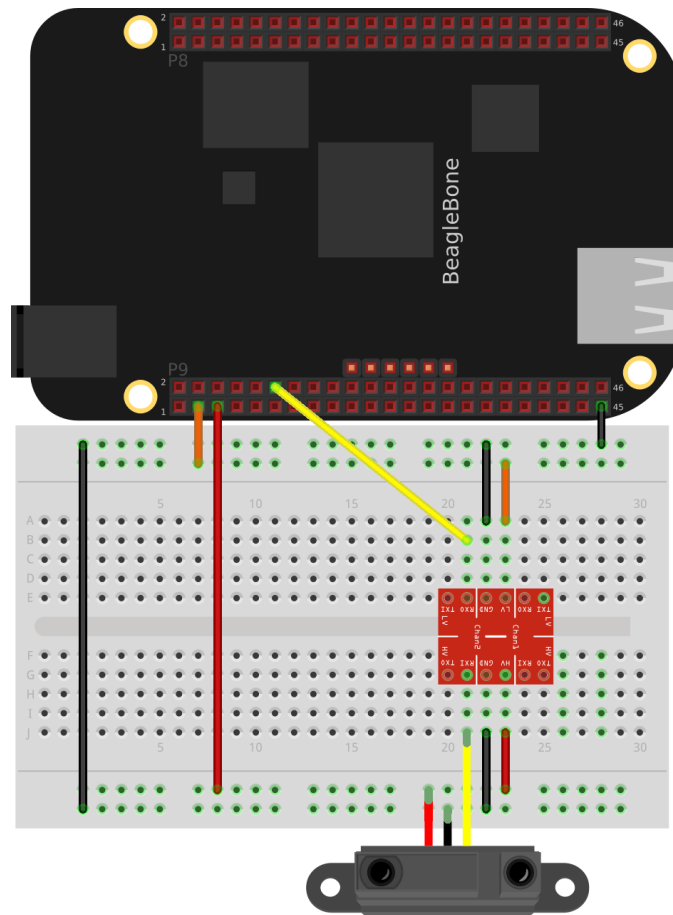


Abbildung 2: Anschluss des Sensors mit einem Pegelwandler [5]

kann dies über eine C++ Library realisiert werden. Dies ist aber für den Vergleich der Sensorik nicht von Belang.

Zum Testen der Entfernung wurde vor den Sensor in einer gewissen Entfernung ein Blatt Papier gehalten.

```

1 | var b = require('bonescript');
2 | b.pinMode('P8_19', b.INPUT);
3 | b.digitalRead('P9_12', printStatus);
4 | function printStatus(x) {
5 |   console.log('x.value = ' + x.value);
6 |   console.log('x.err = ' + x.err);
7 | }

```

Listing 1: Abfragen des Sensor Wertes mit BoneScript [3]

2.1.5 Bewertung

Der Sensor lässt eine sehr einfache Ansteuerung über einen digitalen Pin des BeagleBone Black zu. Dies ist aber auch gleichzeitig ein Nachteil, da die Entfernung nicht im Quellcode/ zur Laufzeit angepasst werden kann, sondern nur per Hardware Stellrädchen. Damit ist zum Beispiel dynamisches Anpassen der Entfernung im Quellcode oder eine Konfigurierung per Remote Interface leider nicht möglich.

Da der zur Verfügung stehende Pegelwandler 4 Kanäle hat, könnten an einem Pegelwandler bis zu 4 Sensoren angeschlossen werden. Das Gewicht bzw. die Größe eines

Pegelwandlers kann in Anbetracht der Sensoren/ des BeagleBone Black vernachlässigt werden.

Damit eignet sich dieser Sensor für eine grundlegende Hinderniserkennung, hat aber klare Grenzen.

2.2 SHARP GP2D12

Die SHARP Sensor Reihe wird im Hause der HAW Hamburg bereits in mehreren Projekten, wie zum Beispiel Projekt “Carolo-Cup” verwendet. Da man so auf Wissen Anderer im Haus zurückgreifen kann und er auf Lager ist, bietet es sich an, ihn ebenfalls für dieses Projekt zu nutzen. Ausgewählt für diesen Zweck wurde ein Sensor mit einer Reichweite von 10cm bis 80cm, da dies am Besten den erwarteten Entfernungen unseres Roboters entspricht.

2.2.1 Technische Daten

- Analoger Output, entfernungsabhängig, siehe Abbildung 3 (S. 6)
- Operationsspannung: 5V
- Reichweite: 10cm bis 80cm

[1]

2.2.2 Anschluss an analogen Pin

Da der analoge Eingang des BeagleBone Black bis zu 1.8V verträgt und der Sensor bis zu 2.8V ausgibt, wird ein Spannungsteiler benötigt, um den Eingangsspannung des BeagleBone nicht zu zerstören. Um die Rechnung zu vereinfachen wird der Innenwiderstand des BeagleBone Black vernachlässigt und damit kann von einem unbelasteten Spannungsteiler ausgegangen werden.

$$U_{R1} = U_{gesamt} - U_{R2}$$

$$U_{R1} = 2.8V - 1.8V = 1.0V$$

$$\frac{U_{R2}}{U_{R1}} = \frac{R2}{R1}$$

$$\frac{1.8V}{1.0V} = \frac{R2}{R1}$$

$$\frac{1.8V}{1.0V} = \frac{9}{5}$$

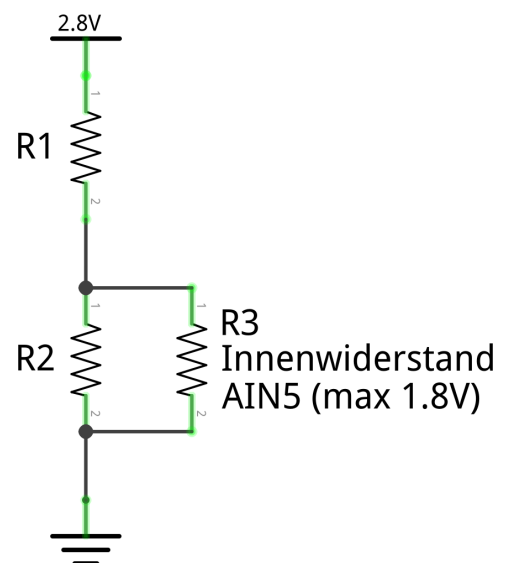


Abbildung 4: belasteter Spannungsteiler [5]

Während der Experimentierzeit waren mehrere Widerstände in Griffreichweite. Darunter befanden sich mehrere 220Ω Widerstände sowieso einige jeweils nur einmal vorhandene, größere Widerstände. Keine der Widerstandskombinationen entsprach weder direkt noch in Reihe verschaltet dem 9:5 Verhältnis. Es müssen also Widerstände parallel geschaltet werden, um das gewünschte Widerstandsverhältnis von $R1$ und $R2$ zu erreichen.

Dabei kann man sich auf die Grundannahmen in 5.1.1 “Parallelschaltung von Widerständen” (S. 12) stützen.

Wenn man für $R1$ und $R2$ im Spannungsteiler jeweils einen 220Ω Widerstand parallel mit einem anderen Teilwiderstand schaltet, werden die jeweiligen Widerstände $R1$ und

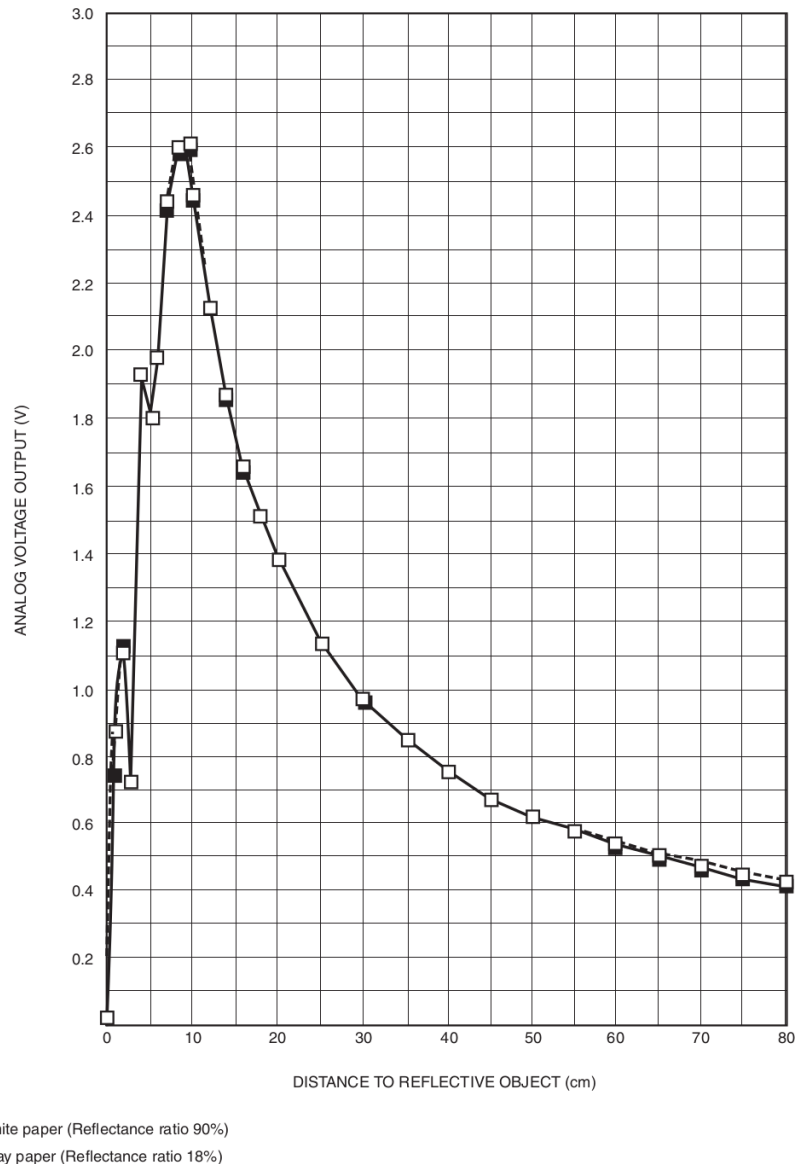


Abbildung 3: Verhältnis Spannung/ Distanz [1]

R_2 jeweils kleiner als 220Ω sein. Da R_1 auf Grund des Verhältnisses 9:5 kleiner sein muss als R_2 (Spannungsteiler: Am kleinsten Widerstand fällt am wenigsten Spannung ab), muss zum 220Ω Teilwiderstand in R_1 ein kleinerer Widerstand parallel dazu geschaltet werden als in R_2 . R_1 besteht somit aus zwei parallelen 220Ω Widerständen, so ist $R_1 = R_{11} || R_{12} = 220\Omega || 220\Omega = 110\Omega$.

Den Widerstand R_2 kann man nun über das gewünschte Widerstandsverhältnis berechnen.

$$R_2 = \frac{R_1}{5} * 9 = \frac{110\Omega}{5} * 9 = 198\Omega$$

$$\frac{1}{R2} = \frac{1}{R21} + \frac{1}{R22}$$

$$\frac{1}{198\Omega} = \frac{1}{R21} + \frac{1}{220\Omega}$$

$$\frac{1}{198\Omega} - \frac{1}{220\Omega} = \frac{1}{R21}$$

$$\frac{1}{1980\Omega} = \frac{1}{R21}$$

$$1980\Omega = R21$$

Der Widerstand der 1980Ω am besten entsprach, war ein 1.5kΩ Widerstand.

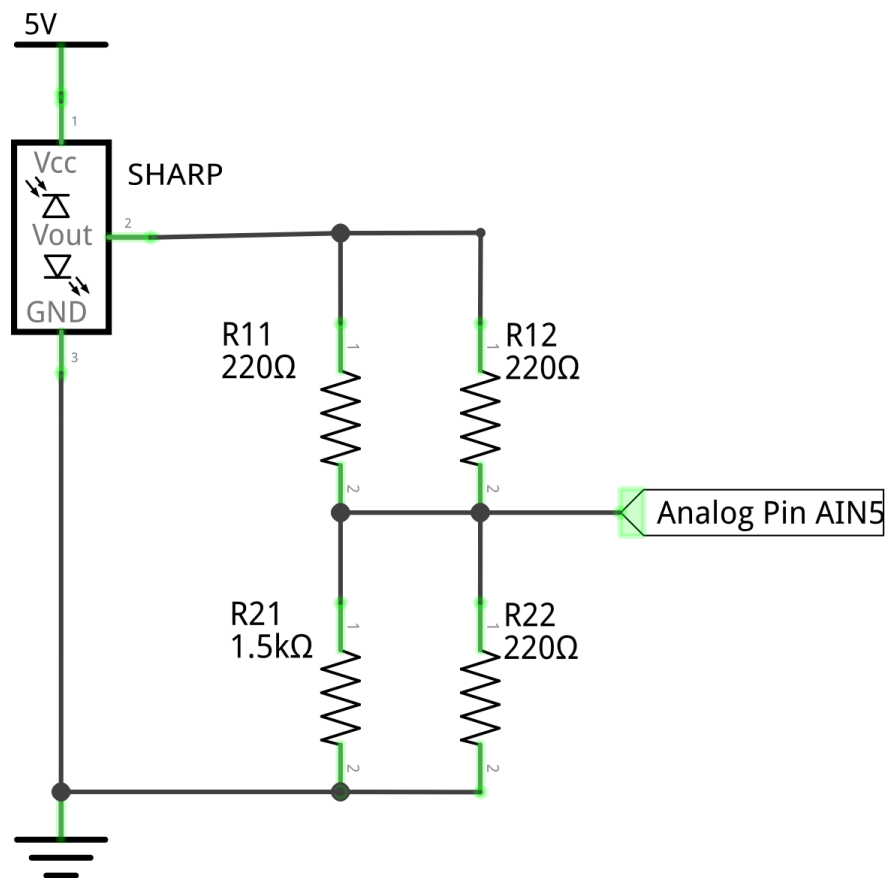


Abbildung 5: Schaltplan Anschluss SHARP Sensor mit einem Spannungsteiler [5]

Der Spannungsteiler mit den hier gewählten Widerständen ist in Abbildung 5 (S. 7) zu sehen.

Um einen Rechenfehler zu vermeiden, der die Hardware zerstören könnte, wird noch eine Kontrollrechnung durchgeführt, die sicherstellt, dass die maximale Spannung am analogen Eingangspin 1.8V nicht übersteigt:

$$U_{Pin} = \frac{R2}{R1 + R2} * U_{Sensor}$$

$$R1 = R11 || R12 = 220\Omega || 220\Omega = 110\Omega$$

$$R2 = R21 || R22 = 1500\Omega || 220\Omega \approx 192\Omega$$

$$U_{Pin} = \frac{192\Omega}{110\Omega + 192\Omega} * 2.8V \approx 1.78V$$

Daraus folgt, dass die Spannung am analogen Eingangspin die 1.8V nicht überschreiten wird.

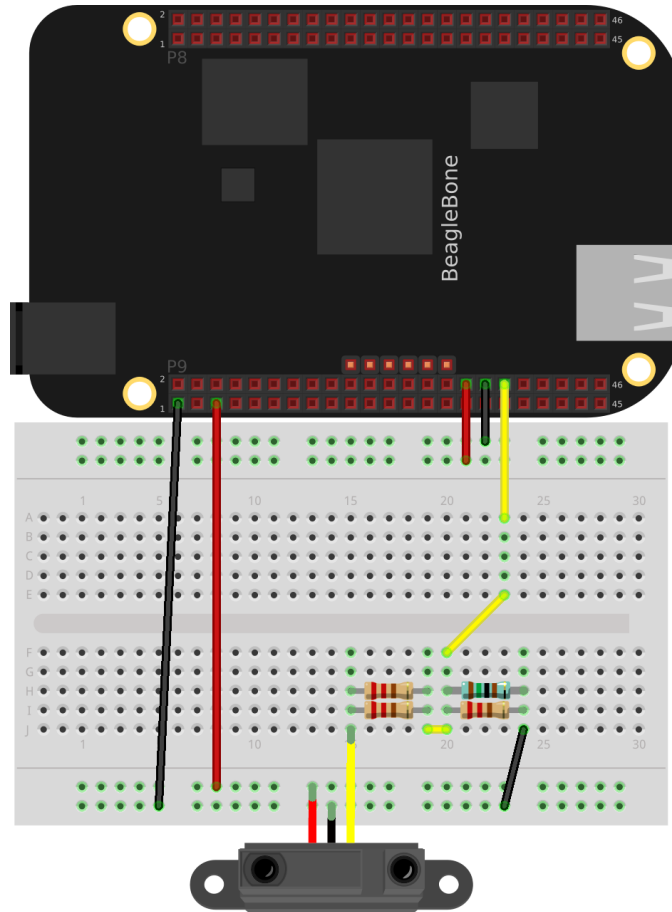


Abbildung 6: Anschluss des SHARP Sensors [5]

Die Beschaltung der Hardware ist in Abbildung 6 (S. 8) zu sehen.

Wie bereits eingangs erwähnt wurde der Innenwiderstand des BeagleBone Black Analog Pins vernachlässigt. Da dieser parallel zu R_2 liegt, wird er diesen verkleinern. Die Spannung, die an diesem abfällt, ist also ebenfalls geringer und wird die 1.78V nicht annähernd erreichen.

2.2.3 Ansteuerung per BeagleBone Black

Um die gemessene Entfernung des Sensors abzufragen, muss zunächst der analoge Wert vom Pin (P9_36) eingelesen werden. Für den ersten Test erfolgt dies wieder mit BoneScript, wie in Listing 2 (S. 8) zu sehen.

```

1 | var b = require('bonescript');
2 | b.analogRead('P9_36', printStatus);
3 | function printStatus(x) {
4 |   console.log('x.value = ' + x.value);
5 |   console.log('x.err = ' + x.err);
6 | }

```

Listing 2: Abfragen des analogen Pins mit BoneScript [4]

In Abbildung 3 (S. 6) kann man das nicht lineare Verhältnis zwischen Spannung und gemessener Distanz des Sensors sehen. Um die gemessene Entfernung zu ermitteln könnte man das nicht lineare Verhältnis in Quellcode abbilden. Eine simple Alternative dazu ist die Messgenauigkeit, wie sie auch in ADC zu Einsatz kommt. So teilt man die Spannungsebenen in Abschnitte ein und weist ihnen einen Entfernungswert zu. Diese Art des Einlesens könnte als ein zusätzliches Feature auch per Software zur Laufzeit (nach-) kalibriert werden.

2.2.4 Bewertung

Der Sensor lässt durch ein analoges Signal eine genauere Abstandserkennung zu. Der Benutzer einer C++ Sensor Klasse kann so später selbst entscheiden, wie er mit der gemessenen Entfernung umgehen will und ist damit sehr frei in seiner Benutzung. So kann zur Laufzeit ein Sensor nachkalibriert werden oder dynamisch auf Entfernungen reagiert werden.

Für den späteren Einbau im Roboter werden zusätzlich nur ein paar Widerstände benötigt, die weder vom Platz noch vom Gewicht zur Last fallen werden.

Da der BeagleBone Black nur über 5 analoge Inputs verfügt, ist das Maximum der anschließbaren Sensoren auf 5 begrenzt.

Der SHARP GP2D12 eignet sich gut für Hinderniserkennungen und ist durch die analogen Werte vielfältig in einer späteren Programmierung einsetzbar. Nur durch die wenigen analogen Inputs ist maximale Sensor Anzahl begrenzt.

3 Vergleich der Ansätze

Beide Sensoren im Gegensatz zum BeagleBone Black mit 5V arbeiten, erfordert dies eine Herunterregelung der Spannung. Da der Sensor aus dem ersten Ansatz mit digitalem Output arbeitet, reicht ein einfacher Pegelwandler. Der Sensor aus dem zweiten Ansatz hingegen benötigt einen Spannungsteiler, der aus mehreren Widerständen besteht.

Die Planung mit einem Pegelwandlers ist einfacher, da dieser nur eingebaut werden muss. Im Gegensatz dazu müssen die Widerstände im Spannungsteiler erst dimensioniert werden. Dafür ist der Pegelwandler teurer in der Anschaffung als mehrere Widerstände.

Im Gegensatz zum digitalen Sensor kann man mit dem SHARP Sensor die Entfernungen in der Software dynamischen verwenden. Die Software kann dabei zur Laufzeit auf Entfernungen reagieren, kalibriert werden oder (remote) eingestellt werden.

Der digitale Sensor kann nur über die Potentiometer eingestellt werden. Diese könnten bei einem eingebauten Sensor schwer zugänglich sein.

	Digitaler Sensor	SHARP GP2D12
Output	digital	analog
Reichweite	0cm bis 40cm	10cm bis 80cm
Einstellung Reichweite	Hardware (Potentiometer)	Software (analog Input)
benötigte, zusätzliche Hardware	Pegelwandler	Spannungsteiler

Tabelle 1: Vergleich der Sensoren [1]

4 Zusammenfassung

Es wurden zwei Ansätze der Abstandserkennung mittels Sensoren analysiert und verglichen. Eine simple Lösung ist der digitale Sensor. Er bietet die Grundfunktionalität des Erkennen eines Objektes in Reichweite. Sobald man aber dynamisch zur Laufzeit Anpassungen machen will, kommt man um einen analogen Sensor, wie den SHARP GP2D12, nicht herum.

5 Anhang

5.1 Elektrotechnische Grundannahmen

5.1.1 Parallelschaltung von Widerständen

1. Schaltet man zwei Widerstände parallel, so ist der Gesamtwiderstand immer kleiner als der kleinste Teilwiderstand

Annahme:

$$R2 = x * R1$$

$$x \geq 1$$

zu Zeigen:

$$R1 || R2 = \frac{R1 * R2}{R1 + R2} < R1$$

$$\frac{R1 * R2}{R1 + R2} = \frac{R1 * x * R1}{R1 + x * R1} = \frac{x * R1}{(1 + x) * R1} * R1 = \frac{x}{1 + x} * R1$$

da $x \geq 1$ gilt, gilt auch

$$\frac{x}{1 + x} < 1$$

also

$$\frac{R1 * R2}{R1 + R2} = \frac{x}{1 + x} * R1 < R1$$

2. Schaltet man zwei gleichgroße Widerstände parallel, so halbiert sich der Gesamtwiderstand:

$$R || R = \frac{R * R}{R + R} = \frac{R * R}{2R} = \frac{R}{2} = \frac{1}{2}R$$

5.2 Abbildungsverzeichnis

1	Pegelwandler [2, 5]	2
2	Anschluss des Sensors mit einem Pegelwandler [5]	3
4	belasteter Spannungsteiler [5]	5
3	Verhältnis Spannung/ Distanz [1]	6
5	Schaltplan Anschluss SHARP Sensor mit einem Spannungsteiler [5]	7
6	Anschluss des SHARP Sensors [5]	8

5.3 Tabellenverzeichnis

1	Vergleich der Sensoren [1]	10
---	--------------------------------------	----

5.4 Listings

1	Abfragen des Sensor Wertes mit BoneScript [3]	3
2	Abfragen des analogen Pins mit BoneScript [4]	8

5.5 Quellen

Anschlussbilder und Schaltpläne wurden mit Fritzing [5] unter der CreativeCommon Lizenz erstellt.

Literatur

- [1] SHARP, *GP2D12 Data Sheet*, http://www.sharpsma.com/webfm_send/1203, 2005, aufgerufen am 24.06.2016
- [2] SparkFun, *SparkFun Logic Level Converter - Bi-Directional*, <https://www.sparkfun.com/products/12009>, Jan 2015, abgerufen am 23.07.2016
- [3] BeagleBone BoneScript, *digitalRead*, <http://beagleboard.org/Support/BoneScript/digitalRead/>, 21. Jun 2016, abgerufen am 17.06.2016
- [4] BeagleBone BoneScript, *analogRead*, <http://beagleboard.org/Support/BoneScript/analogRead/>, 19. Jun 2013, abgerufen am 24.06.2016
- [5] Fritzing UG, *Fritzing*, <http://fritzing.org/home/>, Version 0.9.3b, 2. Juni 2016, abgerufen am 18.07.2016