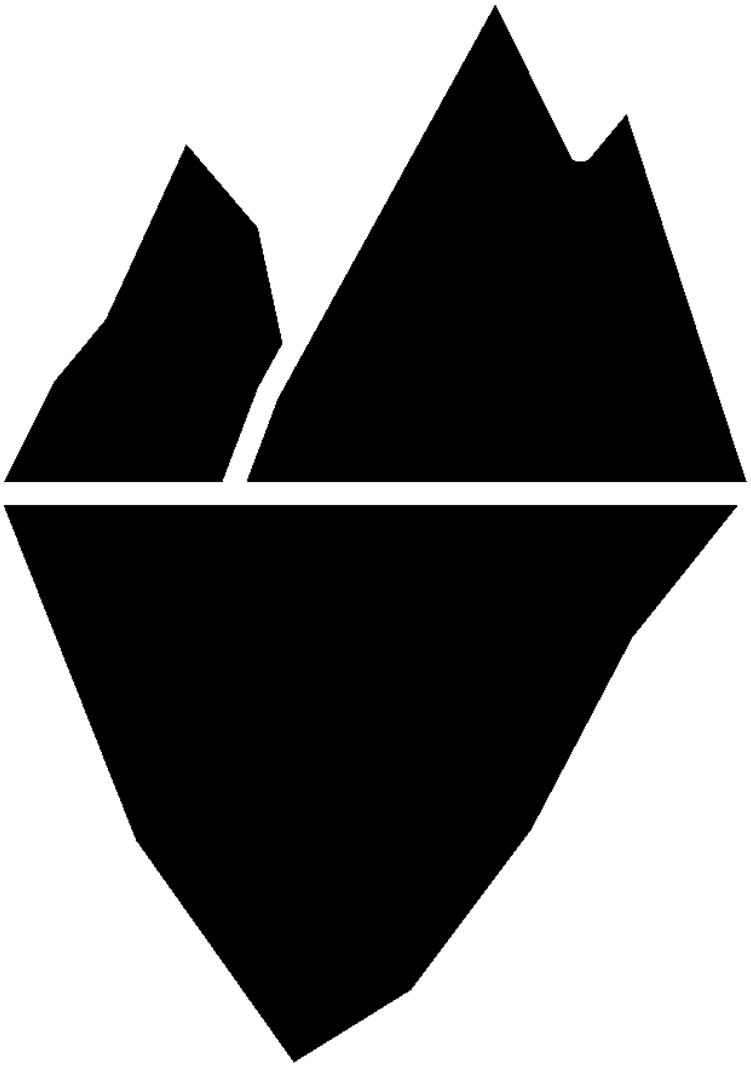

Iceberg Robots

Eine Dokumentation von Oona Kintscher



Seminarkurs Robotik

2018/2019

Inhaltsverzeichnis

1. Projekt im Überblick

Im Sommer 2018 bin ich in das Team Iceberg Robots eingetreten. Unser Team hat an dem RoboCup in der Konkurrenz „2vs2 Soccer“ teilgenommen.

1.1 Wettbewerb - RoboCup

Der RoboCup ist ein internationaler Wettbewerb. Zum ersten Mal ausgetragen wurde er 1997 in Nagoya mit 40 Teams aus der ganzen Welt. Der Wettbewerb ist in zwei Ligen unterteilt: die Major League und die Junior League. Bei dem RoboCup Junior dürfen Teilnehmer mitmachen, die jünger als 19 Jahre sind. Diese Liga ist nochmal unterteilt in drei verschiedene Konkurrenzen, dazu gehören Rescue, OnStage und Soccer. Wie der Name es sagt, spielt man in der Konkurrenz Soccer Fußball. Schülerteams auf der ganzen Welt bauen Roboter, die in dem Wettbewerb gegeneinander antreten. Zuerst startet der Wettbewerb auf regionaler Ebene und man kann sich soweit qualifizieren, dass man bei der Weltmeisterschaft des RoboCups antritt.

1.2 Konkurrenz - Soccer 2vs2 Open

Für die Konkurrenz „2vs2 Soccer“ muss das Team zwei autonom spielende Fußballroboter selbstständig bauen. Diese dürfen das Maximalgewicht von 2400g, den maximalen Durchmesser und die maximale Höhe von 22 cm nicht überschreiten. Darüber hinaus ist jeder Roboter auf eine Kamera und auf eine Betriebsspannung von 15V beschränkt. Bei einem Verstoß gegen die Vorschriften darf der Roboter am Wettkampf nicht teilnehmen. Die Roboter müssen mit einem orangen Ball, der einen Durchmesser von 74 mm (+/- 10 mm) hat, spielen. Das Feld hat eine Größe von 182 cm * 243 cm und besteht aus einer Art Filzteppich. Eine Umrandung in der Höhe von 14 cm grenzt das Spielfeld von der Umgebung ab. Auf dem Spielfeld befindet sich ein Rechteck (122 cm*183 cm), welches durch weiße Linien markiert ist. Dies begrenzt den Raum, in dem die Roboter sich bewegen dürfen. Überschreitet ein Roboter vollständig diese Markierung, wird er für eine Minute oder bis zu dem nächsten Anstoß aus dem Spiel genommen. Außerdem gibt es zwei farbige Tore, die eine Breite von 60 cm aufweisen. Um diese gut unterscheiden zu können, hat das eine Tor eine gelbe und das andere eine blaue Färbung.

Abbildung 1: Abmessungen des Spielfelds

2. Hardware

Dieses Jahr hat unser Team zwei baugleiche Roboter entworfen und gebaut. Im folgenden Texten beschreibe ich den Aufbau dieser Roboter und gehe auf Besonderheiten ein.

2.1 Sensoren

2.1.1 Ultraschallsensor

Die Roboter besitzen fünf Ultraschallsensoren (SRF08), die über eine I2C-Schnittstelle mit unserem Haupt-Mikrocontroller kommunizieren. Zwei der Ultraschallsensoren zeigen nach vorne. Die anderen drei Sensoren zeigen nach hinten, links und rechts vom Roboter. Die Sensoren erfassen den Abstand bis zur nächsten Bande bzw. die Spielfeldbegrenzung. Mit diesem Abstand kann der Roboter sich auf dem Spielfeld orientieren und einordnen. Außerdem kann er durch die zwei vorderen Sensoren ermitteln, ob ein anderer Roboter vor ihm steht oder er sich an der Kante eines Tores befindet. Um zu überprüfen, ob ein anderer Roboter zwischen einer der Banden und dem Roboter steht, bilden wir die Summe der gegenüberliegenden Sensoren und vergleichen sie mit einem fest zugeteilt Wert für die Größe des Spielfelds.

Abbildung 2: Ultraschallsensor SRF08

2.1.2 Kompass

Der 3-Achsen Kompasssensor (BNO055), der auf der Oberseite des Roboters angebracht ist, dient dem Roboter zur Ausrichtung auf das gegnerische Tor. Die Werte des Kompasses werden über die I2C Leitungen zu dem Mikrocontroller übertragen. Zusätzlich können wir mit dem Kompass erkennen, ob der Roboter hochgenommen wurde.

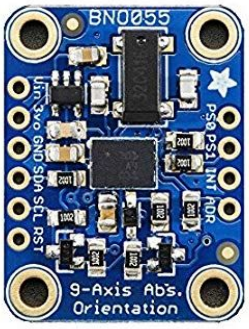


Abbildung 3: Kompass BNO055

2.1.3 Bodensensor

Der Bodensensor besteht bei unserem Roboter aus einer Platine, die mit vielen Fototransistoren und LEDs bestückt ist. Auf der Platine sitzen 40 RGB-Leds, die die Helligkeit der auszuwertenden Bereiche konstant halten. Außerdem besitzt der Bodensensor 41 Fototransistoren, die wir alle mit sogenannten Multiplexer auf 15 Pins minimieren. Somit können wir über drei Pins alle 41 Sensoren auslesen. Die Fototransistoren werden durch einen separaten Arduino Nano ausgewertet, weil die Auswertung parallel zum Hauptprogramm ablaufen soll. Durch einen Interrupt übergeben wir dem Arduino Mega ein Signal, das der Roboter auf einer Linie steht. Die Platine ist auf der Unterseite des Roboters befestigt. Die Anordnung der Transistoren und der RGB-Leds ist speziell. Und zwar gibt es 8 Äste, die in einem

45°-Winkel angeordnet sind und je vier Transistoren besitzen, und einen äußeren Kreis. Jeder Phototransistor hat eine eigene RGB-Led.

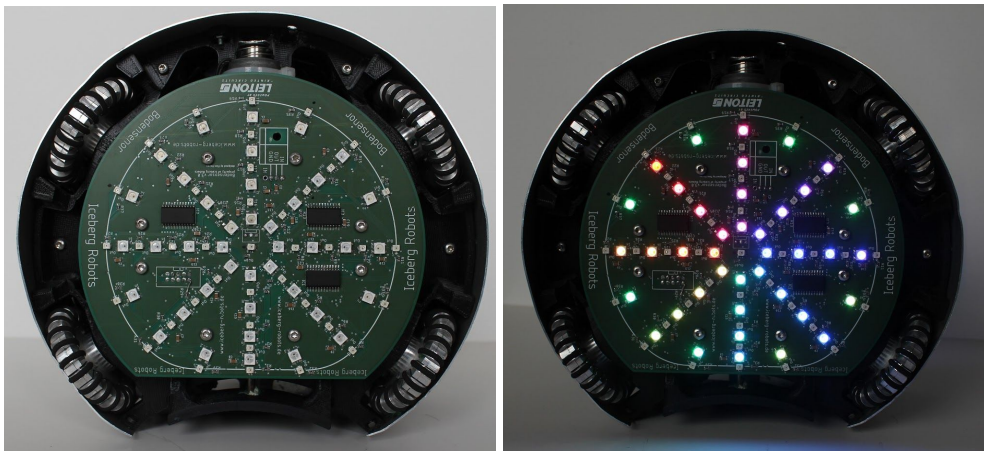


Abbildung 4/5: Anordnung des Bodensensor (mit/ohne Animationen)

2.1.4 Kamera

Um den orangen Ball auf dem Spielfeld zu finden und zu erkennen, benutzen wir eine Kamera (CMUcam5 Pixy)**.** Diese übernimmt die Bildauswertung für uns, weil der Arduino nicht genügend Rechenleistung besitzt, um dies auszuführen. Letztendlich übergibt die Kamera dem Arduino die X- und Y-Koordinaten eines Farbbereiches den wir über eine IDE (PixyMon) festlegen. Dafür nutzt sie eine SPI-Schnittstelle.

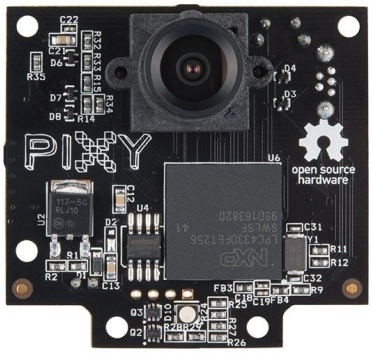


Abbildung 6: Kamera CMUcam5 Pixy

2.1.5 Lichtschranke/Reflexionssensor

Der Roboter besitzt eine Art Lichtschranke, die kann erfassen, ob der Ball direkt in der Ball Capturing Zone des Roboters liegt. Die Lichtschranke besteht aus einer weißen Led und einem Photowiderstand, die nebeneinander, oberhalb der Ball Capturing Zone angebracht sind. Die weiße Led gibt dem auszuwertenden Bereich eine konstante Helligkeit und der Photowiderstand misst das reflektierte Licht. Ist der Ball in der Capturing Zone, dann ist das reflektierte Licht größer als das Licht, wenn der Ball nicht dort liegt. Dieser

Unterschied kommt durch die geringe Lichtabsorption des orangen Balls im Vergleich zum dunkelgrünen Spielfeld.

2.1.6 Encoder

Encoder sind Sensoren, die an den Motoren befestigt sind und die Umdrehung des Motors zählen. Dieses Jahr haben wir es nicht geschafft diese Sensoren in unser Hauptprogramm einzuarbeiten. Allerdings sind sie sehr hilfreich. Bisher war es bei den Robotern der Iceberg Robots nur möglich den Motoren eine definierte Spannung zu geben. Hierbei bedeutet eine hohe Spannung eine hohe Geschwindigkeit und eine geringe Spannung eine niedrige Geschwindigkeit. Mit den Encodern kann man die Umdrehungen der Motoren zählen und somit diese auf eine gezielte Schnelligkeit kontrollieren.

2.2 Aktoren

2.2.1 Motoren

Dieses Jahr besteht die Fortbewegung des Roboters aus vier selbst konfigurierten Motoren der Firma Maxon Motor. Diese bestehen aus dem Planetengetriebe GPX22 C 21:1 und dem Motor DCX22S GB KL12V. Das Planetengetriebe ist eine Keramikversion mit einer Untersetzung von 21:1. Der Motor hat eine Nennspannung von 12 Volt und Graphitbürsten. Zur Ansteuerung der Motoren benutzen wir die Motortreiber L6205N.

2.2.2 OmniWheels

Wie schon die letzten Jahre besitzt der Roboter Omni Wheels. Diese sind Räder, die sich in alle Richtungen bewegen können. Sie bestehen aus mehreren SubWheels, die sich senkrecht zur Drehachse bewegen können. Auf diesem Wege kann der Roboter in alle Richtungen fahren, ohne sich zu drehen. Zusätzlich kann der Roboter so während des Fahrens autonom rotiert werden.

2.2.3 Schuss

Der Schuss an dem Roboter besteht aus einem 3V-Solenoid, einer Achse und einer Platte aus Polylactide. Der Solenoid sitzt an der einen Seite der Achse und die Platte an der anderen Seite. Wird der Schuss durch ein Mosfet ausgelöst, schießt die Achse aus dem Solenoid raus und drückt dabei die Platte mit. Der Ball wird von der Platte angestoßen.

2.3 Kommunikation - Bluetooth

Mit einem Bluetooth-Modul (HC-05), das auf der Hauptplatine sitzt, können die zwei baugleichen Roboter miteinander kommunizieren.

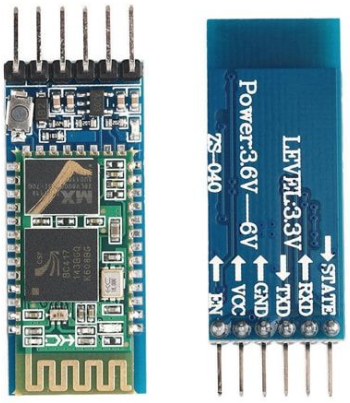


Abbildung 7: Bluetooth-Modul HC-05

2.4 Controller

Als Haupt-Mikrocontroller benutzen wir einen Arduino Mega 2560. Dieser hat 54 digitale Pins, 16 analoge Input Pins und vier UART Pins. Außerdem besitzt der Arduino Mega eine Betriebsspannung von 5V. Der Mega ist zuständig für die Ausführung des Hauptprogramms und somit für das Auslesen aller Sensoren außer dem Bodensensor und der Interpretation der Werte und für die Ansteuerung der Motoren und des Schusses. Additional haben wir noch einen Arduino Nano, der für das Auslesen des Bodensensors verantwortlich ist, und zwei AT-Megas, die die Encoder auslesen sollen.

2.5 Platinen & 3D-Drucke

Dieses Jahr hat das Team drei Platinen selbst entwickelt und herstellen lassen. Diese Platinen setzen sich aus der Bodenplatine (siehe Bodensensor), der Hauptplatine und der PUI. Die Pui ist ein Physical User Interface. Sie dient als Schnittstelle zwischen dem Arduino und uns, dem Benutzer. Durch viele Eingabemöglichkeiten, wie Knöpfe, Switches, einem Rotary Encoder und einem Poti, können wir dem Programm Anweisungen geben. Außerdem gibt es RGB-Leds, die uns den Zustand bestimmter Werte und boolescher Variablen, mit Farben anzeigen können. Durch einen 16 Bit I/O Expander erweitern wir unsere digitalen Pins um weitere 14 und sprechen diese mit einer I2C Schnittstelle an. Die Hauptplatine bietet Platz für alle Sensoren außer dem Bodensensor. Zusätzlich befinden sich fünf Sicherungen auf der Platine, damit nicht durch zu hohe Stromstärken bzw. Kurzschlüsse der ganzen Roboter zerstört wird. Der Hauptschalter, der den Arduino und allen anderen Bauteile mit dem Lipo

verbindet, befindet sich auch auf der Hauptplatine. Ein sehr hilfreiches Feature sind zwei grüne SMD Leds, die einen Heartbeat anzeigen. Andere Features, wie Lautsprecher und Buzzer, sind auf der Platine natürlich auch enthalten. Um ein wenig Spielraum zu lassen sind noch weiter Anschlüsse für UART und I2C auf die Platine gesetzt worden. 3D-Drucke halten die Platinen und andere Bauteil am richtigen Platz. Darüber hinaus grenzt ein Blech den Roboter ab. Dieses dient zum Schutz des Roboters.

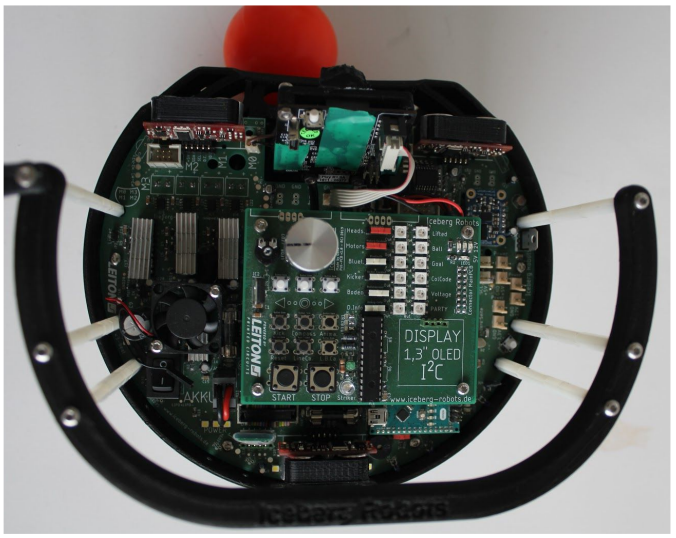


Abbildung 8: Roboter von oben

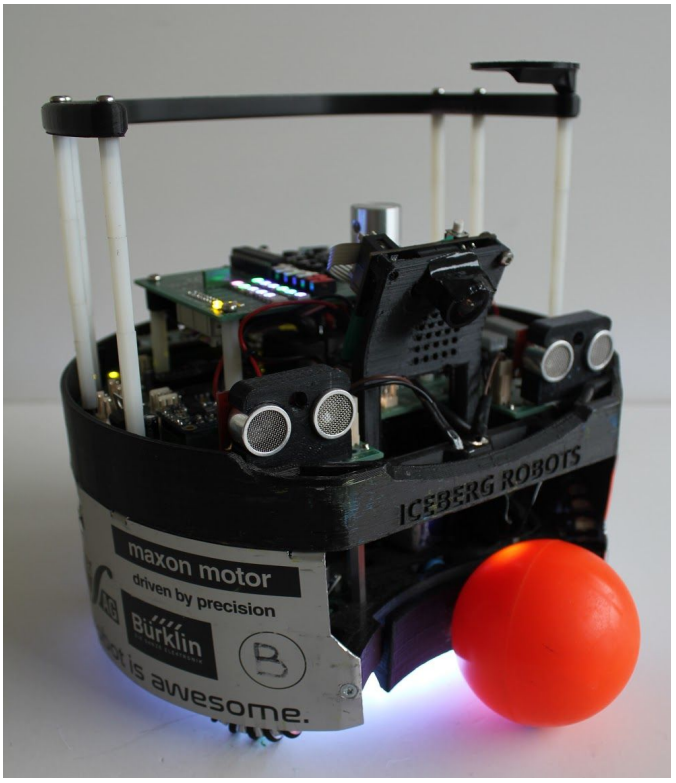


Abbildung 9: Roboter von der Seite

3. Software

3.1 Hauptprogramm

Das Hauptprogramm wird auf dem Arduino Mega ausgeführt. Das Programm wird auf der IDE Visual Studio Code geschrieben und durch GitHub für alle zugänglich gemacht. Obendrein hilft GitHub bei der Strukturierung der Änderung des Programms. Durch unvorhergesehene Komplikationen, die sich im Laufe der Saison im Team ergeben haben, mussten wir kurzzeitig auf das Programm des letzten Jahres zurückgreifen. Dieses Programm wurde größtenteils von Anton Pusch geschrieben. Um das Programm auf dem neu gebauten Roboter abspielen zu können, mussten wir natürlich mehrere Werte und Pins anpassen. Letztendlich war unser Team in Software und Hardware eingeteilt. Mein Bereich war die Hardware. Aus diesem Grund habe ich bei der Programmierung nicht viel mitgearbeitet habe.

3.2 Linienauswertungsprogramm

Das ist das Programm, dass auf dem Arduino Nano läuft und für die Auswertung der 41 Fototransistoren auf dem Bodensensor ist. Sie sollen den Winkel der Linien ermitteln und letztendlich an den Arduino Mega schicken.
Wir haben acht Äste mit Fototransistoren. Jeder dieser Äste ist ein bestimmter Winkel zugeordnet. Am Ende wird der Durchschnitt der ausgeschlagene Äste berechnet und das bildet den Einschlagwinkel der Linie. Das Problem liegt bei der Berechnung des Einschlagwinkels einer Linie, die bei dem Ast mit dem Winkel 0° und dem Ast mit dem Winkel 315° anschlägt. Wird der Durchschnitt dieser Winkel genommen, kommt man auf einen Einschlagwinkel mit rund 158°. Das stimmt jedoch nicht, weil der Einschlagwinkel zwischen 315° und 360° liegen sollte. Dafür haben wir einen Algorithmus geschrieben, um dieses Problem zu lösen.
Zuerst werden die Werte der einzelnen Fototransistoren mit einem Schwellenwert verglichen. Alle Fototransistoren deren Werte unter diesem Schwellenwert liegen werden in einem Array als true gekennzeichnet. Dieser Array wird dann bei der Berechnung des Einschlagwinkels der Linie benutzt. Als Erstes werden die schon genutzten Winkel-, Äste- und Lücken-Arrays auf ihre Ausgangswerte zurückgesetzt. Außerdem wird in den acht Ästen überprüft welcher Sensor mit kleinstem Abstand zum Mittelpunkt ausgeschlagen ist. Dieser wird dann in einen Array geschrieben (Phase 1).
Das Programm soll den größten Bereich finden, in dem kein Phototransistor ausgeschlagen hat. Dies finden wir heraus in dem wir den Ast-Array nutzen, um zu überprüfen, ob einer der Phototransistor in dem bestimmten Ast etwas gesehen hat. Letztendlich wird der Startpunkt der größten Lücke als „maxValue” gespeichert (Phase 2).

Im Anschluss wird der Sprung verschoben, in dem wir auf alle Winkel der Äste 360° addieren bis wir zum Anfang der Lücke kommen. Jetzt berechnen wir den Durchschnitt aller Winkel der ausgeschlagenen Äste. Somit sollte das Problem der Lücke/des Sprungs behoben sein. Zuletzt rechnen wir dann doch Modulo 360, damit wir keinen Winkel über 360° haben (Phase 3).

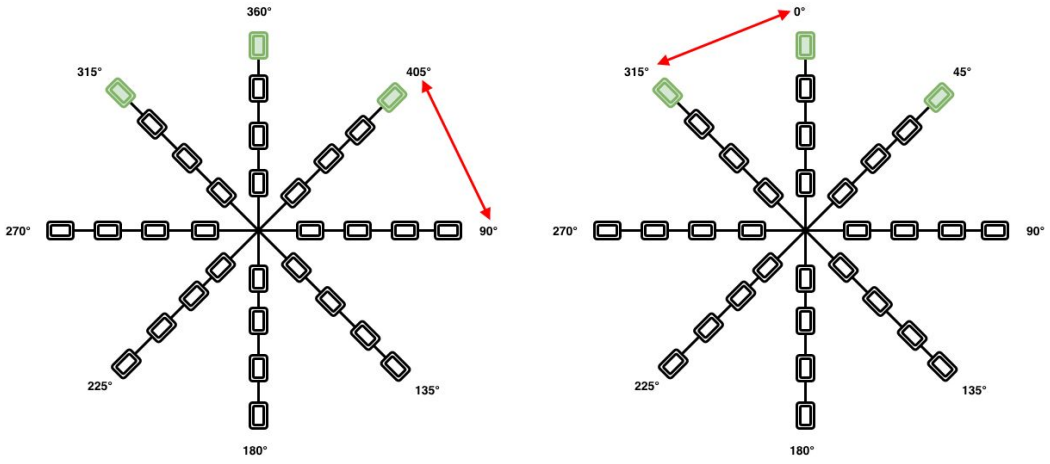


Abbildung 10: Verschiebung der Lücke

```
//angles zurücksetzen
for(int i = 0; i<8; i++){
  angles[i] %= 360;
}

power = 0;
angle = -1;
if (!line)
  return;
int sum = 0;
int count = 0;

//Branches zurücksetzen
for (int i = 0; i < 8; i++)
  branches[i] = 0;

//Branchwerte berechnen
for (int b = 0; b < 8; b++) {
  for (int i = 3; i >= 0; i--) {
    if (hit[b * 4 + i])
      branches[b] =  4 - i;
  }
}

//Gaps zuruecksetzen
for(int i=0; i<8; i++){
  gaps[i] = 0;
}
```

Abbildung 11: Phase 1 der Linienauswertung

```
//groesste Gap ermitteln
byte maxVal = 0;
for(int i=0; i<16; i++){
  if(branches[i%8] == 0){
    gaps[i%8]++;
    if(i != 0){
      gaps[i%8] += gaps[(i-1)%8];
    }
    if(gaps[i%8] > gaps[maxVal]){
      maxVal = i%8;
    }
  }
}
```

Abbildung 12: Phase 2 der Linienauswertung

```
//Wertesprung zu Gap verschieben
for(int i = 8; i > maxVal; i--){
  angles[i%8] += 360;
}


//Durchschnitt bilden
for(int b = 0; b < 8; b++){
  count += branches[b];
  sum += branches[b] * angles[b];
}
angle = sum / count;

angle %= 360;
```

Abbildung 13: Phase 3 der Linienauswertung

4. Quellen

Abbildung1:
https://www.robocupgermanopen.de/sites/default/files/soccer_rules_final_2019.pdf
Abbildung 2:
https://www.antratek.de/srf08-high-performance-range-finder
Abbildung 3:
https://www.amazon.de/Adafruit-Absolute-Orientation-Fusion-Breakout/dp/B06Y1WJB5K
Abbildung 6:
https://tienda.bricogeek.com/descatalogado/755-camara-pixy-cmucam5.html
Abbildung 7:
https://www.fingerpointengg.com/product/bluetooth-module-hc-05/