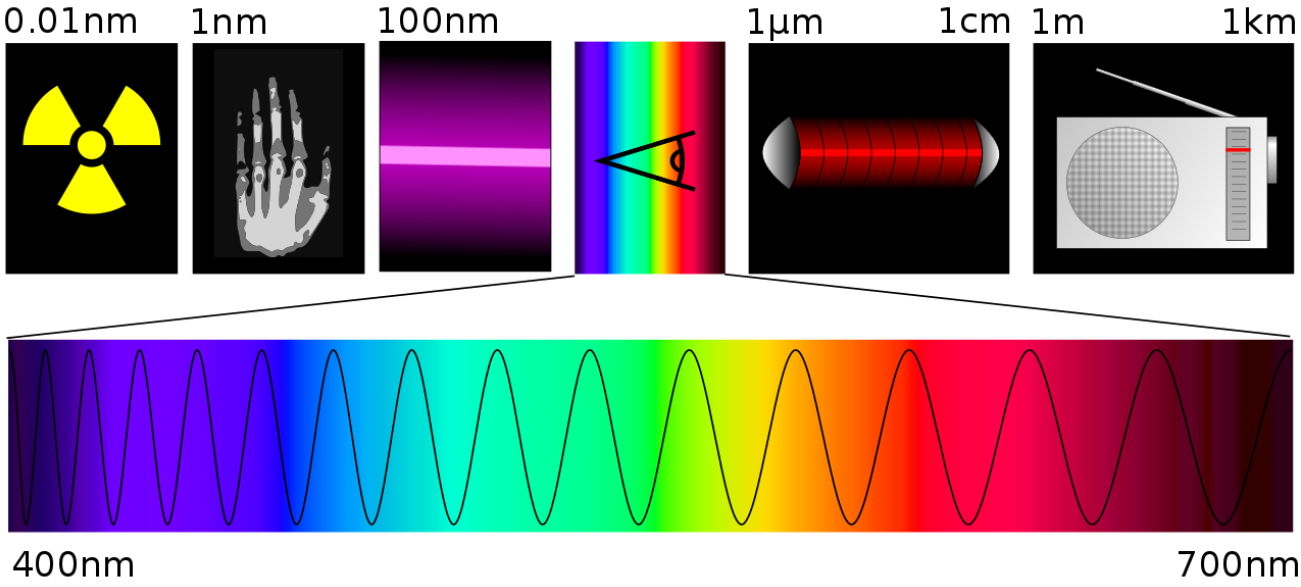
Licht ist eine elektromagnetische Strahlung, die unser Auge sehen kann: unser Auge ist ein „Licht-Sensor“!

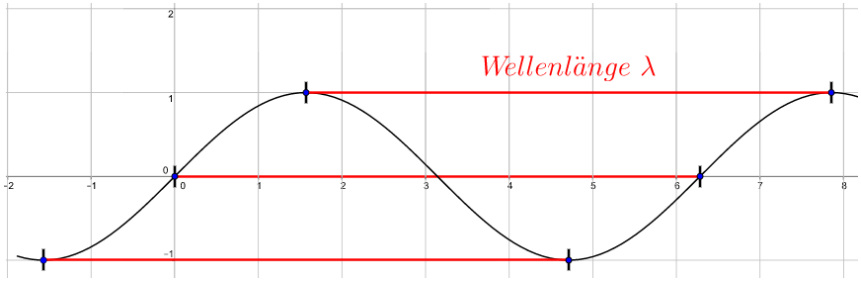
Es gibt verschiedene elektromagnetische Strahlungen, die unterschiedliche Eigenschaften haben:

Ganz links im Bild radioaktive Strahlung, dann Röntgen und ultraviolette (UV-)Strahlung. Alle 3 können uns Menschen gefährlich werden (UV-Strahlung verursacht z.B. Sonnenbrand).

In der Mitte des Bildes sieht man den Bereich der elektromagnetischen Strahlung, die wir „Licht“ nennen. Es gibt hier verschiedene Farben: von blau zu rot.



Farben kann man durch ihre Wellenlänge unterscheiden: das sind die Abstände zwischen den Wellen, also z.B. zwischen 2 Wellenbergen. Man gibt die Wellenlänge in Meter an. Licht hat aber sehr kurze Wellenlängen: 700nm bedeutet „Nano-Meter“ und das sind 0,0007mm (kleiner als $\frac{1}{1000}$ Millimeter)!



Blaues Licht hat eine kleine, rotes Licht eine größere Wellenlänge.

Wenn die Wellenlänge noch größer als rotes Licht wird, dann kann unser Auge es nicht mehr sehen. Diese Wellen nennt man „Infra-rot“ (das heißt: „über-rot“). Mit Infrarot-Strahlung kann man Wärme übertragen (z.B. Infrarot-Heizung oder -Sauna). Wenn die Strahlung nicht extrem stark ist, kann sie uns Menschen **nicht schädlich werden – gleiches gilt für Funk- und Handywellen**!

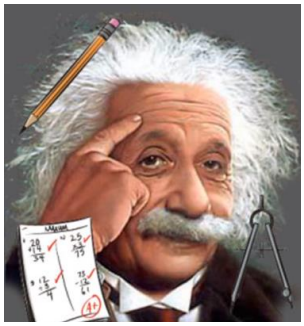
Wenn die Wellen noch länger werden, dann kommt man in den Bereich der Radio- bzw. Funkwellen. Deren Wellenlänge reicht vom Zentimeter- bis zum Kilometer-Bereich.

Die Wellenlänge kann man auch in eine Frequenz umrechnen. Frequenz bedeutet die Anzahl der Wellen pro Sekunde; ihre Einheit ist eigentlich „pro Sekunde“. Zu Ehren des Entdeckers der elektromagnetischen Wellen nennt man diese Einheit „Hertz“ (man schreibt Hz).

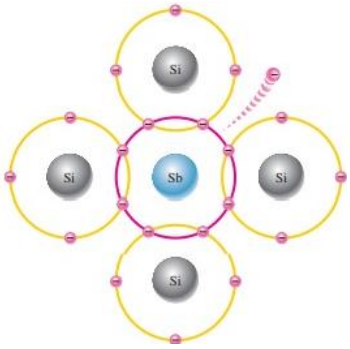
Zentimeter-Wellen haben eine Frequenz im Gigahertz-Bereich (1GHz sind 1000 Millionen Schwingungen pro Sekunde) und werden von Handys mit kleinen Antennen empfangen und ausgesendet. Radio-Wellen sind zwischen 1 Meter lang (z.B. Ö3 auf 89,2 Mega-Hertz oder kurz 89,2MHz – deshalb sind die einfachen Radio-Antennen auch circa einen Meter lang) und können bis zu 10 Kilometer lang werden (Langwelle mit 100.000 Schwingungen pro Sekunde = 100 Kilo-Hertz oder kurz 100kHz).

Licht-Sensoren:

Wie kann ein Sensor Licht messen? Und welche Eigenschaften des Lichts kann er eigentlich messen?



Viele elektronische Bauteile bestehen aus einem interessanten Material: aus Halbleiter. Bei diesen kann elektromagnetische Strahlung - also auch Licht - Elektronen vom Atom loslösen. Dadurch stehen mehr Elektronen als ohne Licht für den Stromtransport zur Verfügung -> je mehr Licht, desto mehr Strom fließt.



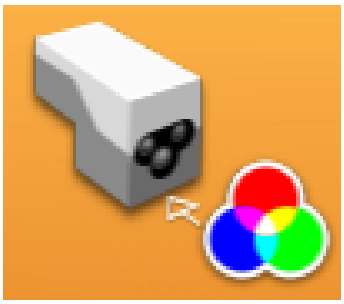
Albert Einstein hat für diese Entdeckung des „Photoelektrischen Effekts“ vor 99 Jahren den Nobelpreis bekommen.

Es kommt auf das Halbleiter-Material an (es gibt verschiedene wie z.B. Silizium, Germanium, usw.), bis zu welcher längsten Wellenlänge dieser Effekt funktioniert (nicht alle reagieren auf Infra-rot, aber alle auf sichtbares Licht, UV- und Röntgenstrahlung.

Will man Farben mit einem Sensor erkennen, muss man mehrere Lichtsensoren mit unterschiedlichen Farbfiltern verwenden – weil der Sensor selbst keine Farben auseinanderhalten kann.

Oder man beleuchtet den Gegenstand mit unterschiedlichen Farben, wie beim Lego Mindstorms EV3 oder Boost: der bunte Gegenstand wird immer nur mit einer Farbe beleuchtet und je nach der Farbe des Gegenstands wird mehr oder weniger Licht reflektiert. Die Farben der Beleuchtung werden nach jeder Messung ganz schnell umgeschaltet und erst durch die vielen Messungen kann der Sensor abschätzen, was das für eine Farbe ist. Man erkennt beim Sensor 2 Linsen: hinter der großen Linse befinden sich die gefärbten LEDs, die das Licht aussenden, und hinter der kleinen Linse ein Lichtempfänger (Fotodetektor).

Manchmal erkennt der Sensor die Farben nicht richtig und verwechselt z.B. grün mit blau oder braun (zum Beispiel bei Verwendung von bunten Kartons).



Das sehen wir:

Das sieht der Sensor:

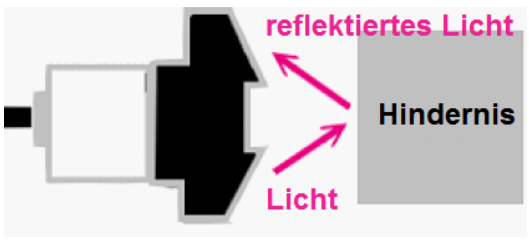
Also eigentlich misst er immer nur Lichtstärken (wie hell oder dunkel ist das Licht), aber für unterschiedliche Farben.

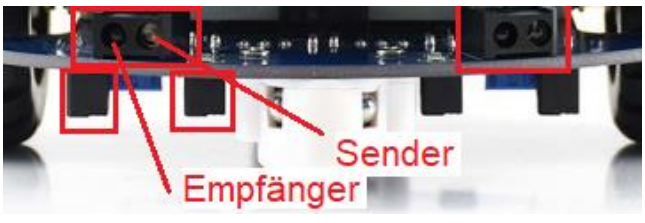
Die meisten Lichtsensoren senden Licht aus und messen das reflektierte Licht. Wenn das Hindernis sehr hell ist (z.B. weiße Farbe an der Wand), wird viel Licht reflektiert und der Empfänger im Sensor zeigt einen hohen Wert an. Wenn das Hindernis dunkel ist, dann ist der Wert niedrig.

mik:robot von vorne:

2 Lichtsensoren für Hindernisse (oben) und

5 Tracking-Sensoren zum Weg-Erkennen (Unterseite)





Leider sind viele Sensoren nicht sehr genau eingestellt (oder hergestellt): das bedeutet, dass sie bei gleichem Licht unterschiedliche Werte anzeigen. Das muss unser Programm korrigieren, indem die Sensoren „kalibriert“ werden. Wenn also der zweite Sensor zu kleine Werte anzeigt, muss das Programm eine Zahl dazuzählen. Oder bei zu großen Werten eine Zahl abziehen.

Es gibt aber noch andere Gründe, warum ein Sensor „kalibriert“ werden muss. Wenn man zum Beispiel mit dem micro:bit erkennen möchte, ob in einem Raum das Licht ein- oder ausgeschaltet ist, dann muss zuerst die Lichtstärke bei ausgeschaltetem Licht und dann die Lichtstärke bei eingeschaltetem Licht gemessen werden.

Und wenn der Raum Fenster hat und es draußen einmal heller und dann wieder dunkler ist, dann muss man nochmals „kalibrieren“.

Wir schreiben nun ein Programm, das beim Drücken des Knopfs A „Licht aus“ und bei Knopf B „Licht ein“ kalibriert. Die 2 unterschiedlichen Werte speichern wir in 2 Variablen.

Jetzt müssen wir im Programm ausrechnen, welcher Wert genau zwischen diesen beiden liegt (Mittelwert):

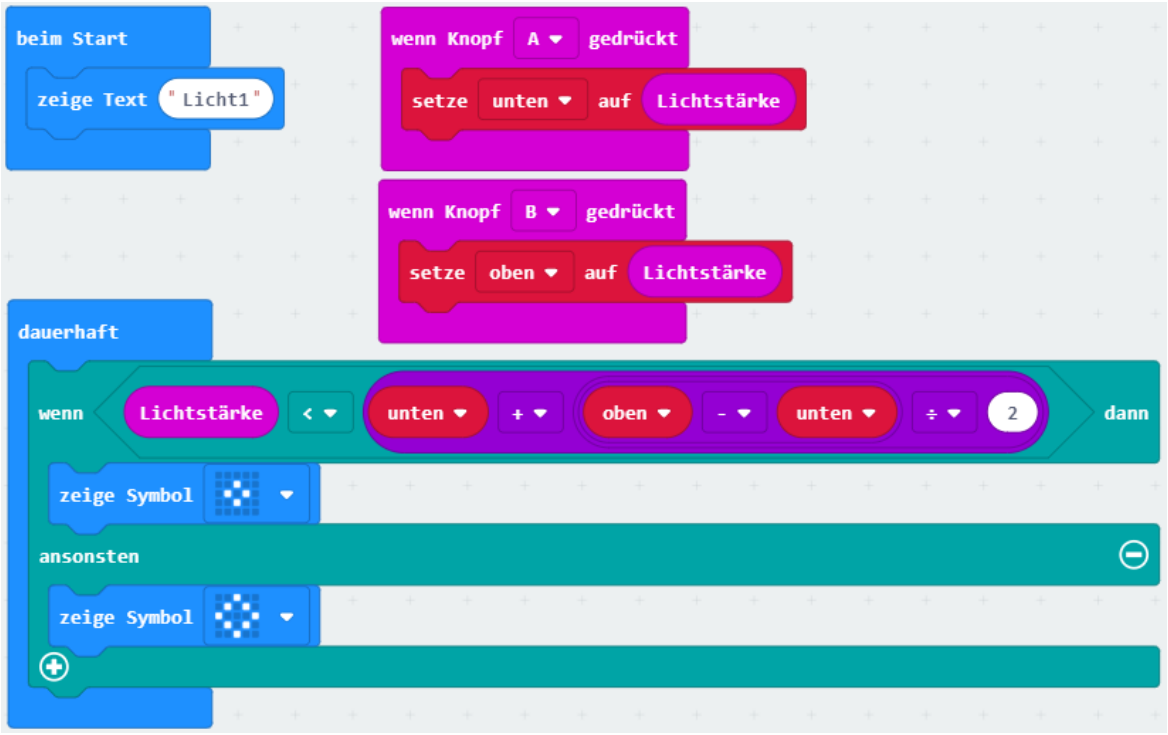
Mittelwert = unten + ((oben – unten) : 2)

Ein Beispiel: unten=20 und oben=30 (die Mitte zwischen 20 und 30 ist 25)

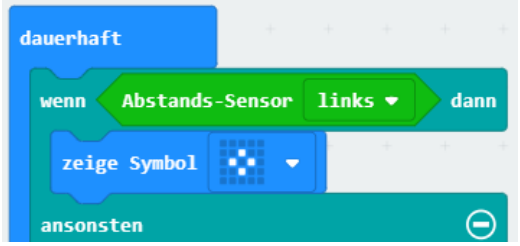
Differenz: oben – unten = 30 – 20 = 10 (10 ist aber nicht die Zahl in der Mitte zwischen 20 und 30)

Man muss die Differenz halbieren (durch 2 dividieren und dann zur unteren Zahl dazuzählen.

Zur Erkennung, ob das Licht ein oder aus ist, vergleichen wir jede Messung mit diesem Mittelwert.



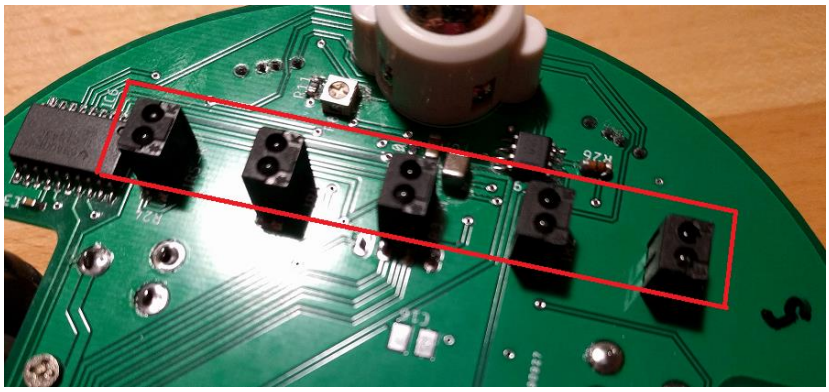
Die Abstands-Sensoren beim mik:robot können auch mit einem Schraubenzieher eingestellt werden (rote Markierungen im Bild) und können in einer Bedingungs-Klammer als „wahr“ oder „falsch“ abgefragt werden:

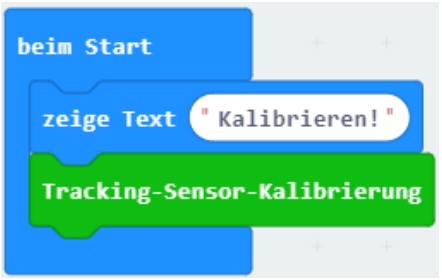


Bei den 5 Sensoren zum Erkennen des Weges auf der Unterseite des Roboters wäre die Kalibrierung mit Knöpfen kompliziert und umständlich. Deshalb gibt es einen Block dafür und der Roboter macht dann eine Drehung nach rechts und nach links.

Es ist wichtig, dass alle 5 Sensoren während des Drehens über die schwarze Linie (wenig Licht-Reflexion) bewegt werden, sonst sind die Werte für „unten“ und „oben“ gleich und die Kalibration funktioniert nicht.

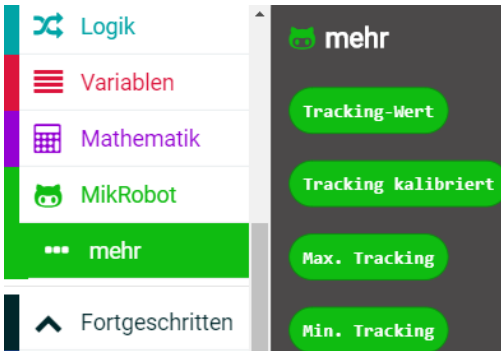
Bei der Kalibration wird für jeden Sensor der unterste und oberste Wert gemessen und es werden Korrekturwerte für jeden Sensor ausgerechnet, damit die Sensoren auf gleichem Untergrund die gleichen Werte liefern.





In der Bibliothek „MikRobot“ gibt es unter „…mehr“ auch alle Werte der Tracking-Sensoren:

1. Tracking-Wert: der aktuelle Messwert jedes der 5 Sensoren (ohne Korrekturzahl)
2. Tracking kalibriert: der aktuelle Messwert jedes der 5 Sensoren mit Korrekturzahl aus der Kalibrierung
3. Max. Tracking: der „obere“ Wert jedes der 5 Sensoren
4. Min. Tracking: der „untere“ Wert



Die Werte der 5 Sensoren sind in einem „Array“ (das ist eine Liste mit 5 Elementen) gespeichert. In Computer-Programmen werden oft Arrays verwendet, weil man damit viele Variablen gleichzeitig bearbeiten kann.

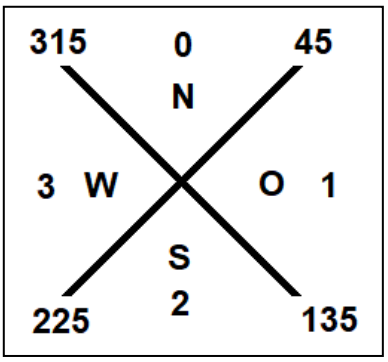
Die Bibliothek „Arrays“ liefert Werte (runde Blöcke) und eckige Blöcke, mit denen man Werte festlegen, neue Elemente hinzufügen oder Elemente aus der Liste löschen kann.

Mit Arrays kann man Programme, wie unseren Kompass aus der ersten Einheit, viel einfacher und übersichtlicher programmieren.

Beim Kompass haben wir vom Sensor einen Winkelwert von 0 bis 360 Grad erhalten und wollten die Himmelsrichtungen „N“, „O“, „S“ und „W“ (West) anzeigen.

Wenn wir eine Liste mit diesen Buchstaben in der richtigen Reihenfolge machen, dann können wir den Sensorwert umrechnen (wir brauchen die Zahlen 0, 1, 2 und 3 für die 4 Elemente).

Der Kompass liefert die folgenden Werte:

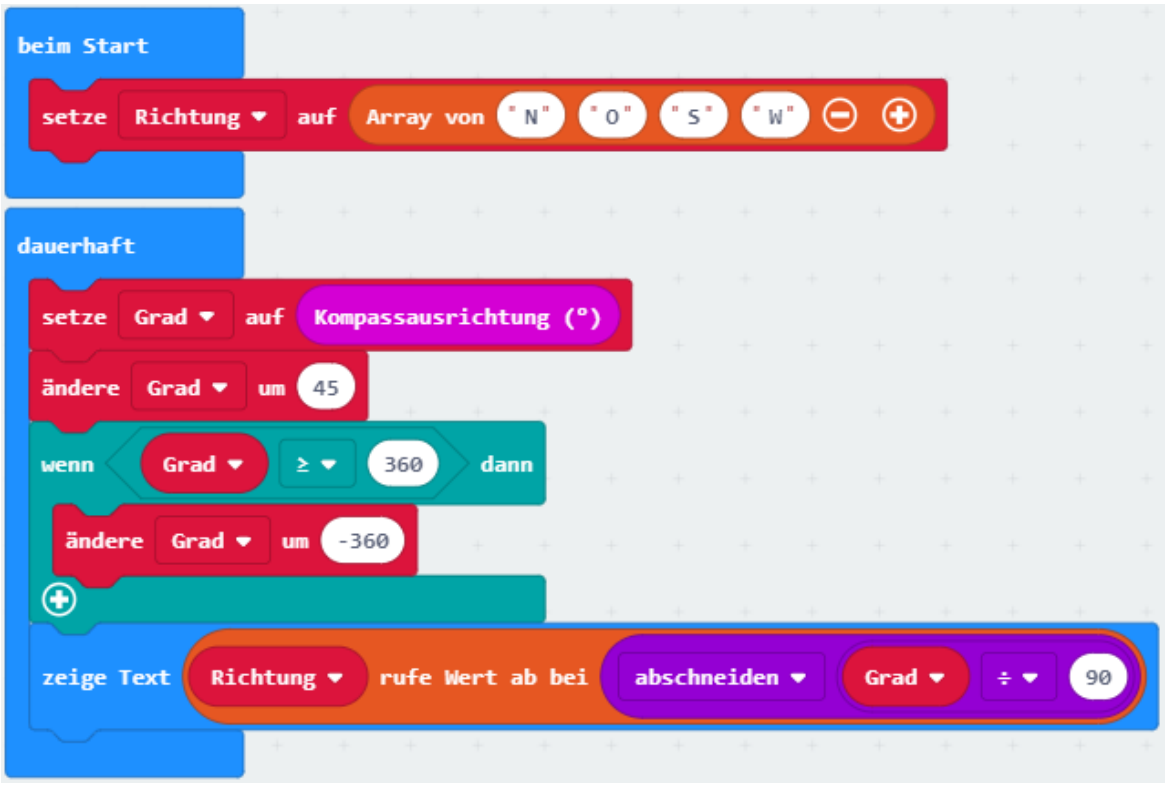


„O“ geht von 45 bis 134 Grad, „S“ von 135 bis 224 Grad usw.

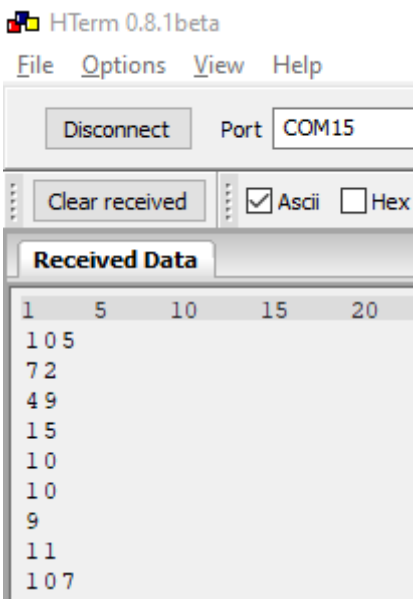
Wenn wir jetzt zum Sensorwert 45 dazuzählen, dann ist „O“ von 90 bis 179 und „S“ von 180 bis 269… Jetzt dividieren wir durch 90: „O“ ist dann eine Zahl zwischen genau 1 und gerade nicht 2 (1,9999…) usw.

Wenn wir beim Ergebnis jetzt noch die Kommastellen wegschneiden, haben wir bereits die richtigen Zahlen für die Elemente.

Nur „N“ passt noch nicht, weil es nach dem dazuzählen von 45 von 45 bis 89 und von 360 bis 404 geht. Dafür brauchen wir noch eine Logik-Abfrage mit einem Vergleich „größer als“:



Oft ist es für einen Programmierer wichtig zu wissen, welche Werte im Programm gerade laufen. Mit der Anzeige auf dem micro:bit kann man nur 1 Buchstaben oder eine einstellige Zahl gut ausgeben. Besser wäre es, wenn man auch lange Zahlen schnell anzeigen könnte.



Dazu kann man Informationen aus dem Programm über das Kabel zum PC schicken. Die Bibliothek dazu heißt „Seriell“.

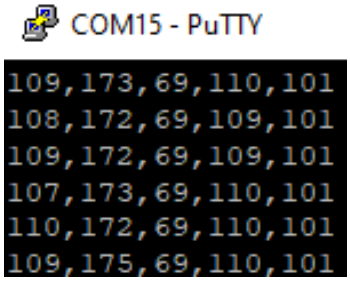
Auf dem PC benötigt man zum Empfangen ein eigenes Programm, das die Profis „Terminal“ oder „Konsole“ nennen. Dieses kann dann Text oder Zahlen übersichtlich und schnell anzeigen.

Auch der Simulator kann eine Konsole anzeigen – probiere es aus!

Das folgende Programm liest mit dem roten Block „setze sensor auf…“ in das Array „sensor“ die aktuellen Werte des Tracking-Sensors.

Dann wird das erste Element (Achtung: die Liste beginnt mit Index 0) über die „serielle“ gesendet, dann ein Beistrich (Text “,“) und dann die nächsten Elemente. Das höchste unserer 5 Elemente (wir haben 5 Sensoren) hat den Index 4.

Wir sehen nun die aktuellen Werte aller Tracking-Sensoren und können überlegen, welcher Wert für eine Entscheidung „schwarz“ (Linie) oder „weiß“ (nicht mehr auf der Linie) gut passt.



Damit können wir ein einfaches Programm für den Roboter schreiben, mit dem dieser einer Linie nachfährt. Wir verwenden dafür nur einen der 5 Sensoren, am Besten den in der Mitte (Index 2). Wenn dessen Wert kleiner als der Entscheidungs-Wert (z.B. 100) ist, dann „sieht“ der Sensor die schwarze Linie. Unser Programm gibt dann auf Motor M1 (links) mehr Gas als auf M2 (rechts) – der Roboter fährt eine Kurve nach rechts.

Damit verliert er aber die schwarze Linie und der Sensor wird einen Wert > 100 liefern, worauf der Roboter rechts mehr Gas gibt und eine Linkskurve fährt. Der Roboter wird also der schwarzen Linie entlang „pendeln“.