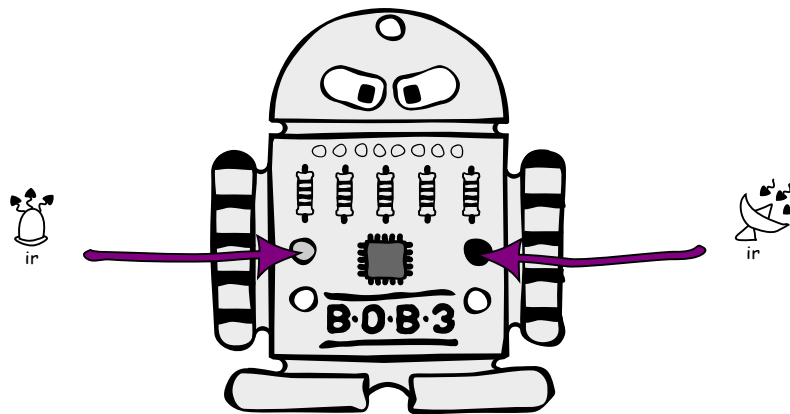


# Arbeitsblatt 14 - IR-Sensor

## BOB3's Infrarotlicht-Sensor



BOB3 hat einen **IR-Sensor**, der aus zwei Teilen besteht: Einer hellblauen **IR-Sende-LED** und einem schwarzen **IR-Empfänger**. Die Abkürzung „IR“ steht für „Infrarot“. Infrarotlicht ist eine spezielle Lichtart. Die IR-Sende-LED **sendet Infrarotlicht** aus und der IR-Empfänger **detektiert das Infrarotlicht**. Mit diesem Sensor kann der Bob **nah** und **fern** unterscheiden, als Lichtschranke arbeiten, bemerken, ob z.B. deine Hand oder ein Blatt Papier vor ihm ist, oder er kann anderen BOB3-Robotern Botschaften senden!

Für diese Features muss Bob nichts berühren, er bemerkt alles komplett berührungslos mittels des **Reflexions-Verfahrens**. Du kannst z.B. ein Spiel programmieren, bei dem die Reaktionsgeschwindigkeit zweier Spieler gegeneinander antritt: Bob gibt Signale und die Spieler müssen möglichst schnell die **Lichtschranke** auslösen. Am Ende rechnet Bob aus, wer schneller war!

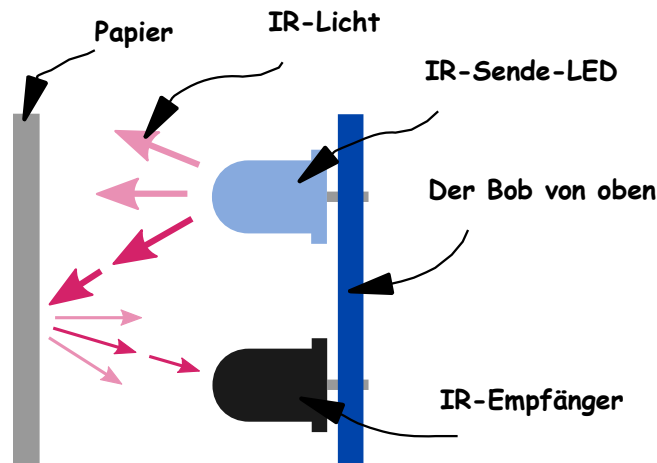
Du kannst Bob auch als **Alarmanlage** programmieren, er kann dann auf Kekse oder andere Dinge aufpassen und schlägt sofort Alarm, sobald er einen Dieb bemerkt!

## Reflexions-Verfahren

Die **Detektion** von **nah** oder **fern** funktioniert nach dem **Reflexionsverfahren**: Die IR-Sende-LED sendet IR-Licht aus, dieses trifft dann auf ein Hindernis (z.B. ein Blatt Papier oder eine Hand), wird von dem Hindernis **zurückreflektiert** und kann so von dem IR-Empfänger empfangen werden.

**Je näher** das Papier vor dem Sensor ist, **desto mehr IR-Licht detektiert** der

IR-Empfänger. Falls das Papier **weiter entfernt** vom Sensor ist, wird **wenig** reflektiertes IR-Licht detektiert.



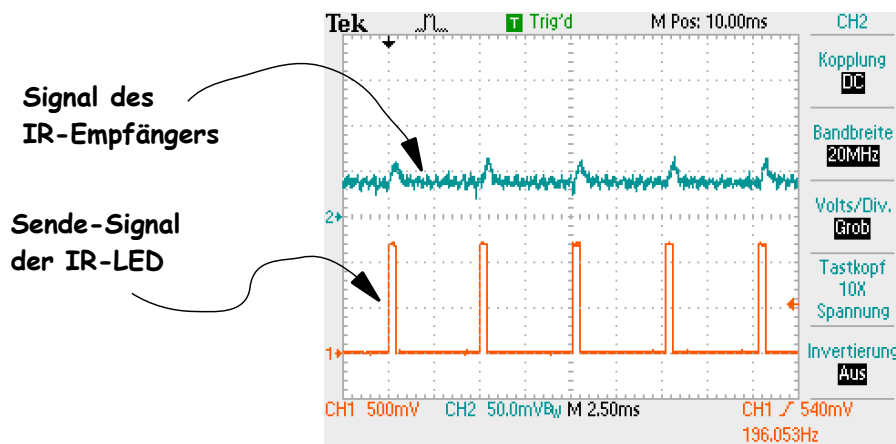
So kann der Bob z.B. auch als Parksensor für eine **Einparkhilfe** programmiert werden:

- Fall 1)** IR-Sensor detektiert kein Hindernis/anderes Auto  
→ LEDs grün, freie Fahrt!
- Fall 2)** IR-Sensor detektiert ein etwas entferntes Hindernis/anderes Auto  
→ LEDs orange, langsamer fahren und aufpassen!
- Fall 3)** IR-Sensor detektiert ein nahes Hindernis/anderes Auto  
→ LEDs rot, STOP!

## Messverfahren

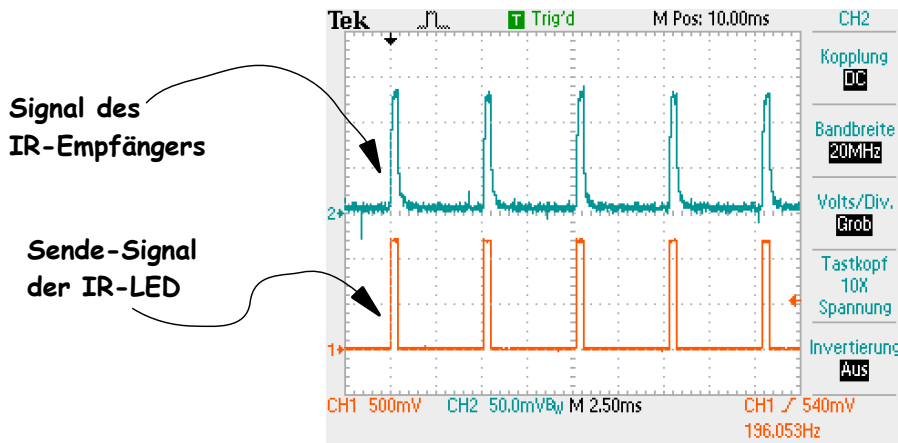
Das Messverfahren lässt sich sehr gut mit einem Oszilloskop veranschaulichen. Hier sieht man als obere Kurve das Signal des IR-Empfängers und als untere Kurve das Sende-Signal der IR-LED.

**Beispiel 1:** - Keine Reflexion, kein Tageslicht -



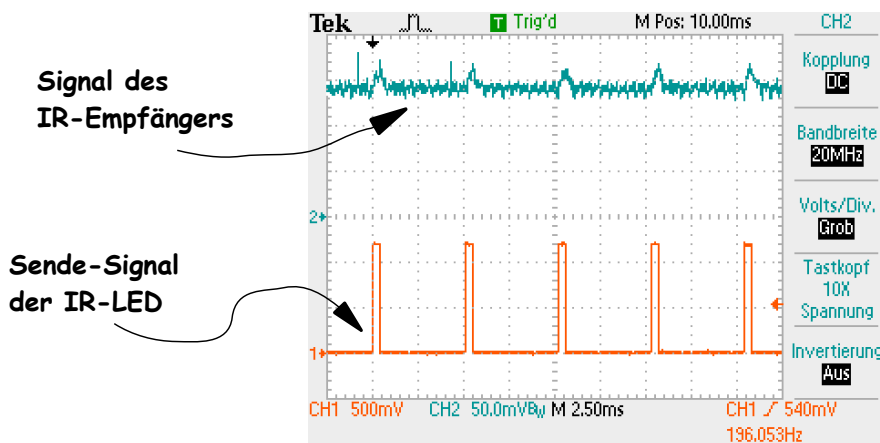
Man sieht an der oberen Kurve deutlich, dass der IR-Empfänger fast keine Signale detektiert.

**Beispiel 2:** - *Kein Tageslicht, Reflexion an einer Hand* -



In diesem Beispiel zeigt die obere Kurve deutliche Ausschläge sobald die IR-LED sendet. Man erkennt, dass der IR-Empfänger Signale von einem reflektierenden Objekt, in diesem Fall von einer Hand, detektiert.

**Beispiel 3:** - *Keine Reflexion, Tageslicht* -



In diesem Beispiel zeigt die obere Kurve fast keine Ausschläge, der IR-Empfänger detektiert also kein reflektierendes Objekt. Deutlich zu sehen ist jedoch, dass die Kurve in der y-Achse verschoben ist. An dem höheren Signalpegel erkennt man deutlich, dass der IR-Empfänger in diesem Fall Tageslicht detektiert.

## Software-Bibliothek

Zur Ansteuerung des IR-Sensors stehen in der Software-Bibliothek des BOB3 fertig implementierte Methoden zur Verfügung:

Methode	Bedeutung	Rückgabewert/Parameter
<code>bob3.getIRSensor()</code>	Liefert den Wert des IR-Reflexions-Sensors	0 - 255
<code>bob3.getIRLight()</code>	Liefert den Wert des IR-Umgebungslichts	0 – 255 255: direktes Sonnenlicht 0: Dunkelheit
<code>bob3.enableIRSensor(enable)</code>	Aktiviert bzw. deaktiviert den IR-Reflexions-Sensor	enable: 1 → aktiviert 0 → deaktiviert
<code>bob3.receiveMessage(timeout)</code>	Empfängt eine IR-Message	0 – 255 / -1 bei keinem Signal timeout: Millisekunden, die gewartet werden sollen
<code>bob3.transmitMessage(message)</code>	Sendet eine IR-Message	Message: 0 – 255, Zahl, die übertragen werden soll

Die Methode `bob3.getIRSensor()` liefert den jeweils aktuellen Wert des **IR-Sensors** als Zahlenwert zwischen 0 und 255 zurück. Falls nichts reflektiert wird, ist der Rückgabewert eine 0, falls ein Objekt IR-Licht reflektiert, dann ist der Rückgabewert eine Zahl zwischen 1 und 255, je nach **Abstand** und **Reflexionseigenschaften** des Objekts.

Die Methode `bob3.getIRLight()` kann sehr gut für **Smart-Home** Beispiele verwendet werden, denn sie liefert den aktuellen Wert des **IR-Umgebungslichts** als Zahlenwert zwischen 0 und 255 zurück. Falls der Sensor direktes Sonnenlicht detektiert, so wird ein hoher Zahlenwert zurückgeliefert. Bei absoluter Dunkelheit wird ein sehr kleiner Zahlenwert bzw. eine 0 zurückgeliefert.

Mittels der Methode `bob3.enableIRSensor(enable)` kann der IR-Sensor aktiviert bzw. deaktiviert werden. Nach dem Aufruf von `bob3.enableIRSensor(0)` ist der Sensor dahingehend deaktiviert, dass die Reflexionsmessung nicht mehr durchgeführt wird. Die passive Messung des Umgebungslichts findet allerdings weiterhin statt.

Mit der Methode `bob3.receiveMessage(timeout)` und der Methode `bob3.transmitMessage(message)` kann ein 8-Bit-Wert von einem Bob zu einem anderen Bob übertragen werden. Die Methode `bob3.receiveMessage(timeout)` bekommt als Parameter die Anzahl an Millisekunden übergeben, die auf eine Nachricht gewartet werden soll. Falls eine gültige Nachricht empfangen wird, liefert die Methode einen Zahlenwert zwischen 0 und 255. Wird innerhalb des gesetzten Zeitraums keine gültige Nachricht empfangen, dann liefert die Methode eine -1 zurück. So können die Roboter sich untereinander **Botschaften** senden, andere Roboter **fernsteuern** oder sich als **Freunde** oder **Feinde** erkennen!

**Aufgabe 1:** Beschreibe, aus welchen Teilen der IR-Sensor von BOB3 besteht und welche Funktion diese Bestandteile haben.

---



---



---



---



---

**Aufgabe 2:** Beschreibe, wie das Reflexions-Verfahren funktioniert.

---



---



---



---



---

**Aufgabe 3:** Welche Programm-Codes sind korrekt und würden compilieren? Kreuze die richtigen Antworten an, es sind mehrere Antworten möglich:

- `bob3.receiveMessage('hallo');`
- `bob3.transmitMessage('hallo');`
- `bob3.transmitMessage(1000000);`
- `bob3.transmitMessage(0);`
- `bob3.transmitMessage(101);`
- `bob3.transmitMessage();`
- `bob3.enableIRSensor(enable);`
- `bob3.enableIRSensor(Enable);`
- `bob3.getIRSensor(0);`
- `bob3.getIRLight(100);`
- `int sensorWert = bob3.getIRSensor();`
- `int sensorWert = bob3.getIRLight();`

**Aufgabe 4:** Beschreibe das folgende Programm und nenne eine Anwendungsmöglichkeit.

```

1 #include <BOB3.h>
2
3 void loop() {
4
5     int sensorwert = bob3.getIRSensor();
6
7     if (sensorwert > 10) {
8         bob3.setEyes(RED, RED);
9     } else {
10        bob3.setEyes(GREEN, GREEN);
11    }
12
13 }
    
```

---



---



---



---



---

**Aufgabe 5:** Beschreibe das folgende Programm und nenne eine Anwendungsmöglichkeit.

```

1 #include <BOB3.h>
2
3 void loop() {
4
5     int sensorwert = bob3.getIRLight();
6
7     if (sensorwert < 10) {
8         bob3.setEyes(WHITE, WHITE);
9         bob3.setWhiteLeds(ON, ON);
10    } else {
11        bob3.setEyes(OFF, OFF);
12        bob3.setWhiteLeds(OFF, OFF);
13    }
14
15 }
    
```

---



---



---



---



---

**Aufgabe 6:** Nenne alle zulässigen Parameter der Methode `bob3.enableIRSensor(enable)`.

---



---