

Die Flugintensität von Bienen

Wie beeinflusst das Wetter ihr Flugverhalten?

schriftliche Arbeit von Benjamin Maaß, 14 Jahre

Erarbeitungsort: Schülerforschungszentrum des Gauß-Gymnasiums Worms

Projektbetreuer: Annette Modl-Chalwatzis, Edgar Gandelheidt

Thema des Projekts: Wie verhält sich die Flugintensität von Bienen

Fachgebiet: Technik

Bundesland: Rheinland-Pfalz

Wettbewerbsjahr: 2022

Kurzfassung

Ich möchte mit meinem Projekt herausfinden, von welchen Faktoren es abhängt, ob Bienen viel oder wenig fliegen. Hierzu entwickle ich in diesem Jahr ein Messinstrument, mit dem ich alle Bienen zählen kann, die den überwachten Eingang passieren. Als Zählmittel werde ich einen Arduino mit einer Lichtschranke nutzen. Das Ergebnis wird anschließend auf einem Laptop angezeigt. Vom angezeigten Ergebnis lässt sich dann ableiten, von welchen Faktoren die Flugfrequenz und damit auch die Honigproduktion beeinflusst wird.

Inhaltsverzeichnis

Kurzfassung	1
Ideenfindung:	2
Aufbau des Arduino:	2
Programmierung des Arduinos	3
Findung des Sensor	9
Eigenschaften des Sensors	9
Aussicht.....	Fehler! Textmarke nicht definiert.
Unterstützer.....	10

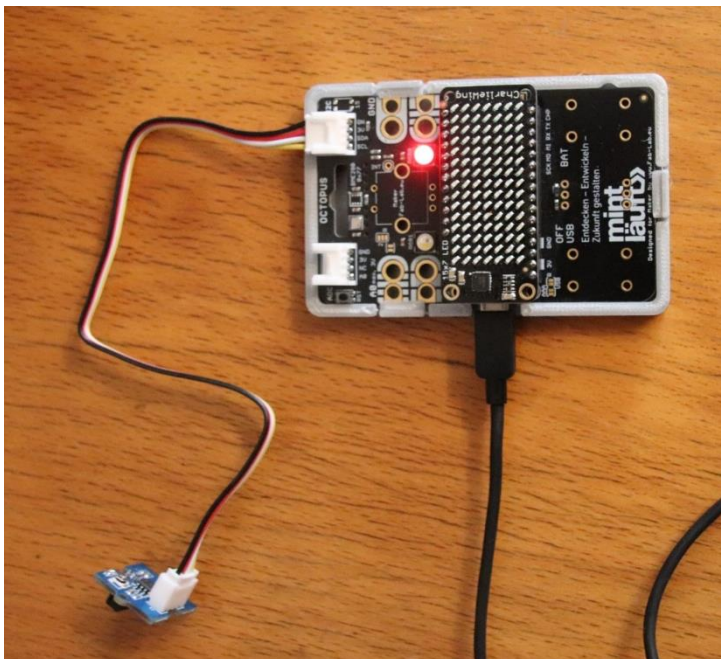
Ideenfindung:

Zur Idee Bienen zu überwachen, kam ich im Mint-Unterricht an unserer Schule. Dort haben wir uns mit Hummeln beschäftigt und dabei besprochen, welche Arten es gibt, was diese mögen und was nicht. Eine Aufgabe war es, einen Bereich an unserer Schule insektengerecht zu gestalten. Außerdem hat uns unsere Lehrerin zur schuleigenen Forscherwerkstatt eingeladen, wo uns der Umgang mit einem Arduino gezeigt wurde. Dabei entwickelte sich die Idee, Bienen zu zählen und herauszufinden, wann und wieviel sie fliegen.

Aufbau des Arduino:

Das Arduino Octopusboard, welches an der Universität Trier und dem Umweltcampus Birkenfeld entwickelt wurde, ist bei meinem Versuchsaufbau das wichtigste Gerät. Der Arduino wurde über die Software Ardu Block programmiert. Nähere Informationen hierzu folgen später.

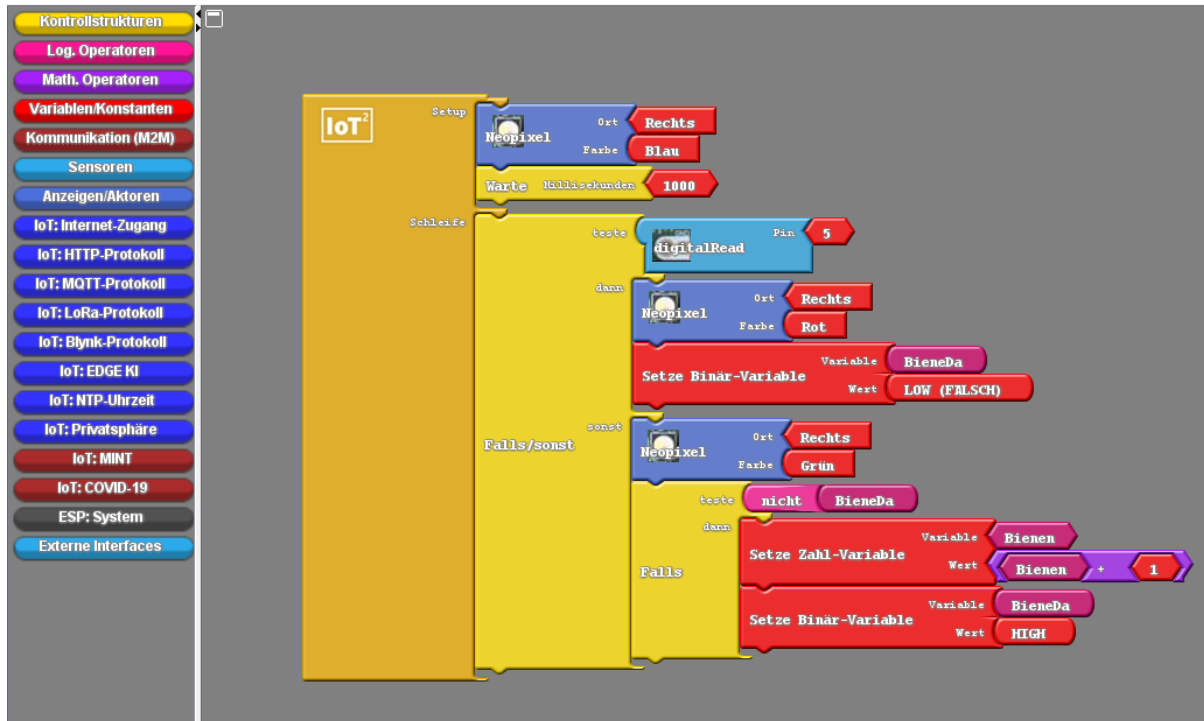
An den Octopus ist eine Lichtschranke mit Hilfe eines Kabels an den Digital Read-Eingang angeschlossen. Über das schwarze Micro USB-Kabel wird der Arduino mit Strom versorgt und die Daten des Sensors an den Seriellmonitor auf den PC übertragen. Der Arduino hat insgesamt zwei mehrfarbige Neopixel. Bei meiner Programmierung wird allerdings nur ein Licht benutzt, welches grün zeigt, sobald der Sensor etwas erkennt und rot zeigt, wenn der Sensor nicht auslöst. Dies dient ausschließlich der zusätzlichen optischen Überwachung, ob die Daten des Seriell Monitors korrekt sind.



Programmierung des Arduinos

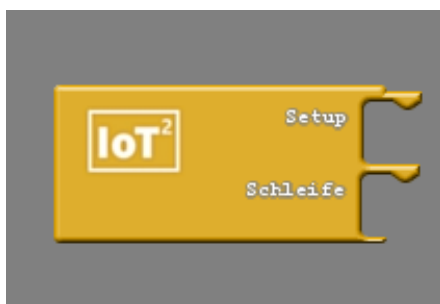
Das Octopusboard wurde mit der grafischen Programmiersoftware Ardublock programmiert.

Das folgende Bild zeigt mein Programmaufbau:



Nun zur Erklärung des Programmes.

Ardublock unterteilt das Programm in zwei Teile: den „Setupbereich“ und den „Schleifebereich“. Der Setupbereich wird beim Start des Arduinos einmalig durchlaufen und hat in meinem Fall keine Auswirkung auf die spätere Zählung. Das Setup dient hier lediglich dazu, beim Hochladen neuer Programmpunkte den Fortschritt des Vorgangs zu überwachen.



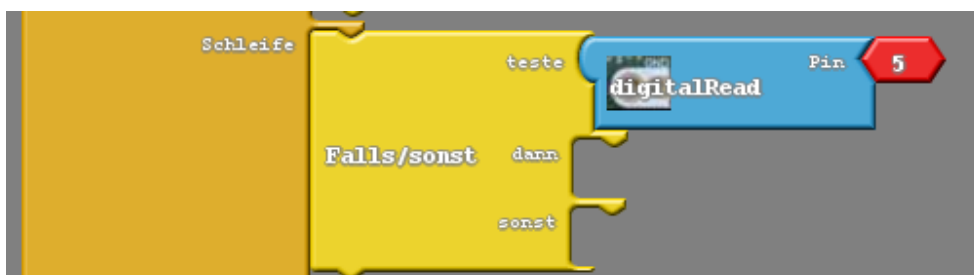
Im Setupbereich habe ich einen Baustein eingefügt, der die rechte „Neopixel“ blau leuchten lässt. Diese wird nach 1000 Millisekunden, also einer Sekunde, durch den gelben Baustein ausgeschaltet. Wäre dieser nicht da, würde die „Neopixel“ nicht ausgehen und der Arduino nicht in den zweiten Teil, den „Schleifebereich“ übergehen.



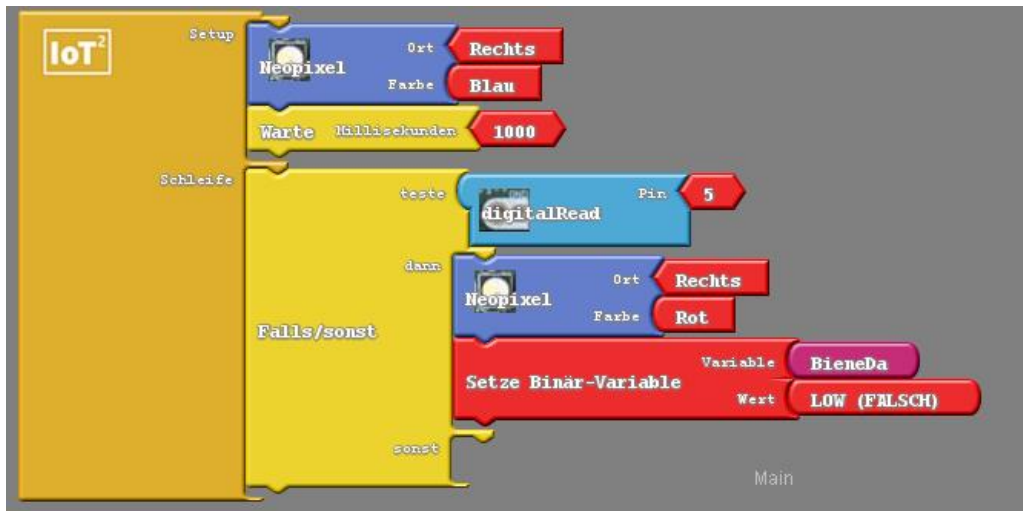
Nun kommen wir zum zweiten Teil, dem „Schleifebereich“, den ich im weiteren Verlauf als „Schleife“ bezeichnen werde. In der „Schleife“ habe ich zunächst eine „Falls/sonst“ Funktion eingefügt.



Diese sorgt dafür, dass in das Feld „teste“ ein Sensorbaustein eingefügt werden kann. In meinem Fall den „digitalRead“-Eingang, der am Octopusboard auf den Pin fünf gelegt wurde. Den Pin kann man an dem kleinen roten Feld eingeben.



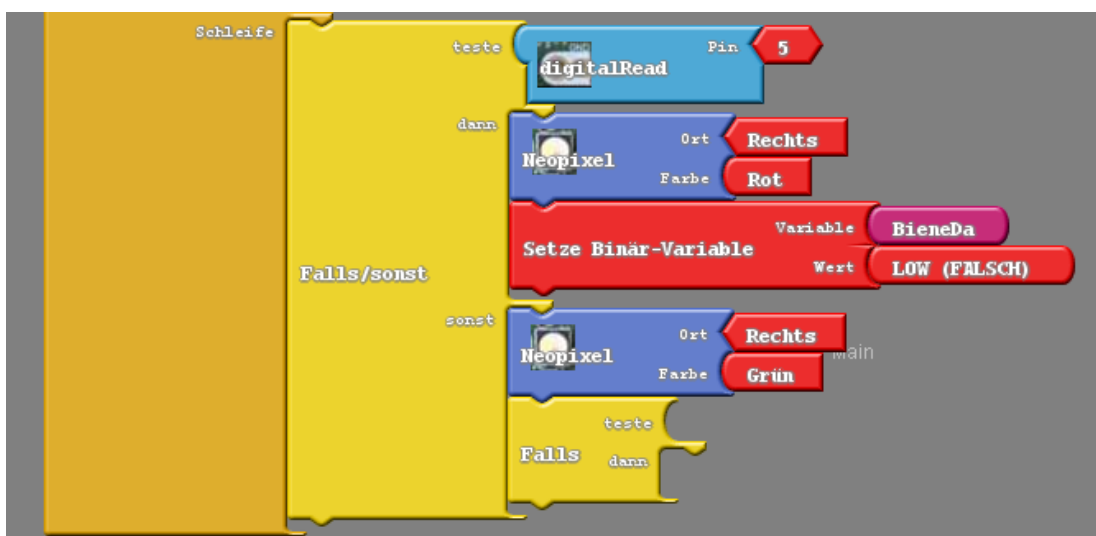
Nach dem Einfügen des Sensors, wird das „dann“ Feld belegt. In dieses Feld wird hinterlegt, was der Arduino machen soll, wenn der Sensor etwas erkennt. Ich habe dieses Feld so befüllt, dass die rechte „Neopixel“ in diesem Fall rot leuchtet und eine „Binär-Variable“ setzt. Diese habe ich „BieneDa“ genannt und den Wert auf „Low“ gesetzt.



Nun habe ich noch den „sonst“-Teil befüllt, in den ich erneut den „Neopixel“- Baustein eingefügt habe. In diesem Fall soll er grün leuchten. Alles was im „sonst“-Teil steht, wird ausgeführt, wenn eine Biene erkannt wird.



Außerdem habe ich in den „Sonst“ Teil noch eine „Falls“-Struktur programmiert.



In der „Falls“-Struktur habe ich den „teste“-Teil ausgefüllt. „nicht“ aus dem Bereich der „logischen Operatoren“ wurde eingefügt und dahinter erneut die Variable „BieneDa“ gehängt. Das „nicht“ sorgt dafür, dass der weitere Teil nur ausgeführt wird, sofern die Variable „BieneDa“ nicht gesetzt ist. Der Teil wird nur ausgeführt, wenn zuvor eine Biene erfasst wurde.

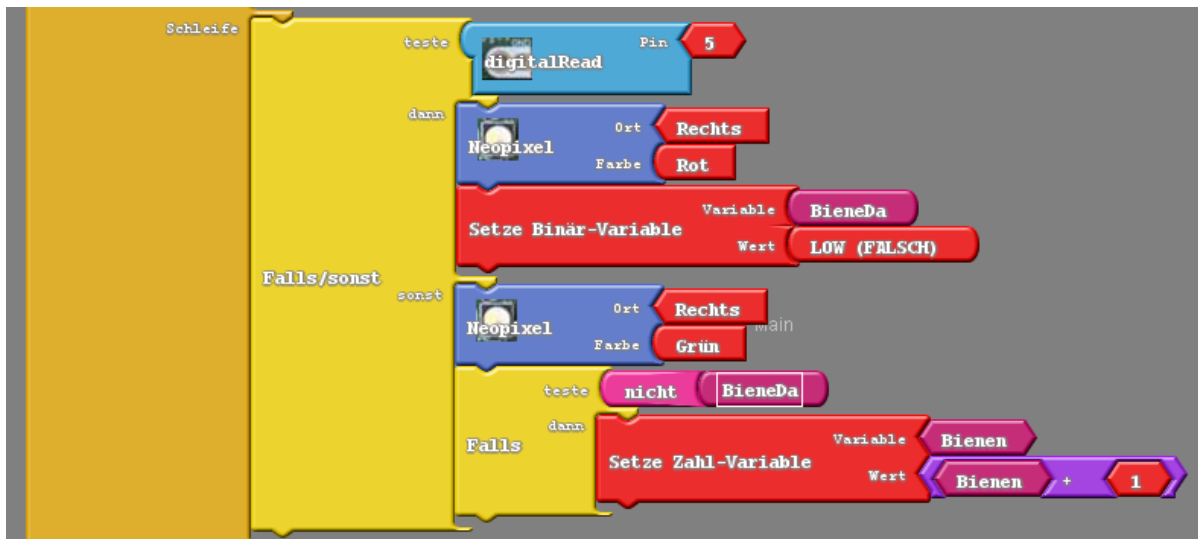


In den „dann“-Teil der „Falls“-Struktur habe ich dann wieder den „Setze Zahl-Variable“-Baustein eingefügt sowie die Variable zu „Bienen“ umbenannt.

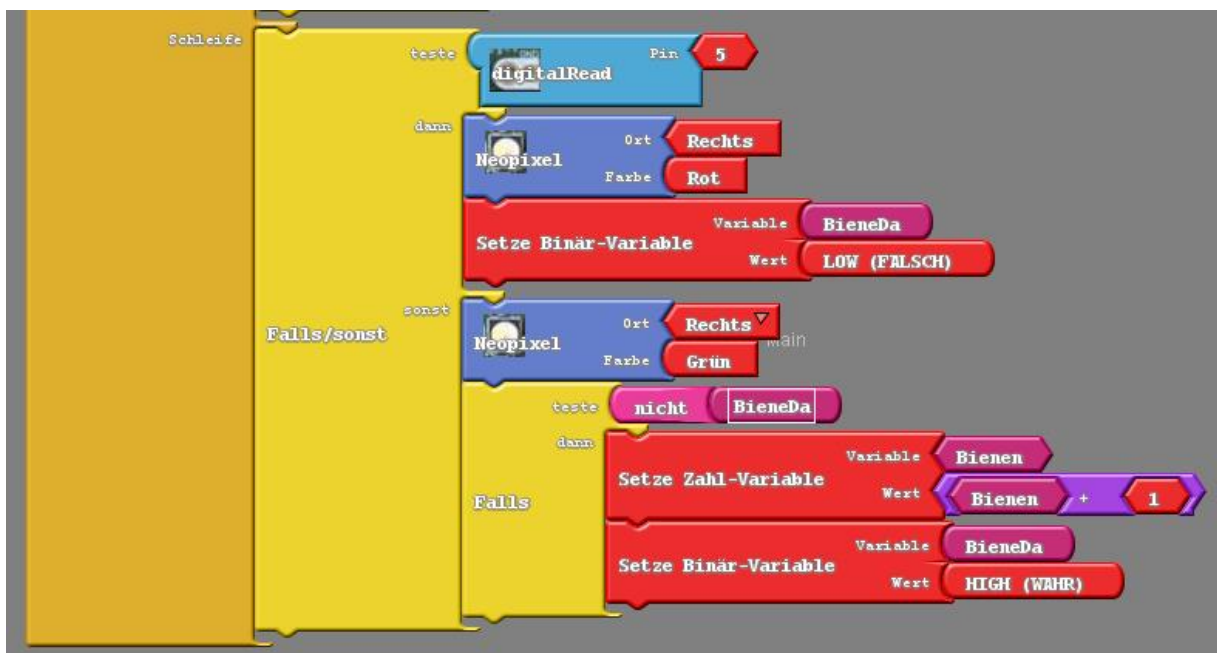


Anschließend änderte ich den Wert von 0 zu 1 und setzte ihn in den zweiten Teil der Plusklammer. In den ersten Teil der Plusklammer fügte ich die Variable „Bienen“ ein. Der komplette hier beschriebene „Setze Zahl-Variable“-Baustein sorgt dafür, dass die Variable -

also in der späteren Anwendung die Bienen - um eins nach jedem Unterbrechen der Lichtschranke erhöht wird. Das Ergebnis kann man später im „Seriell Monitor“ sehen.



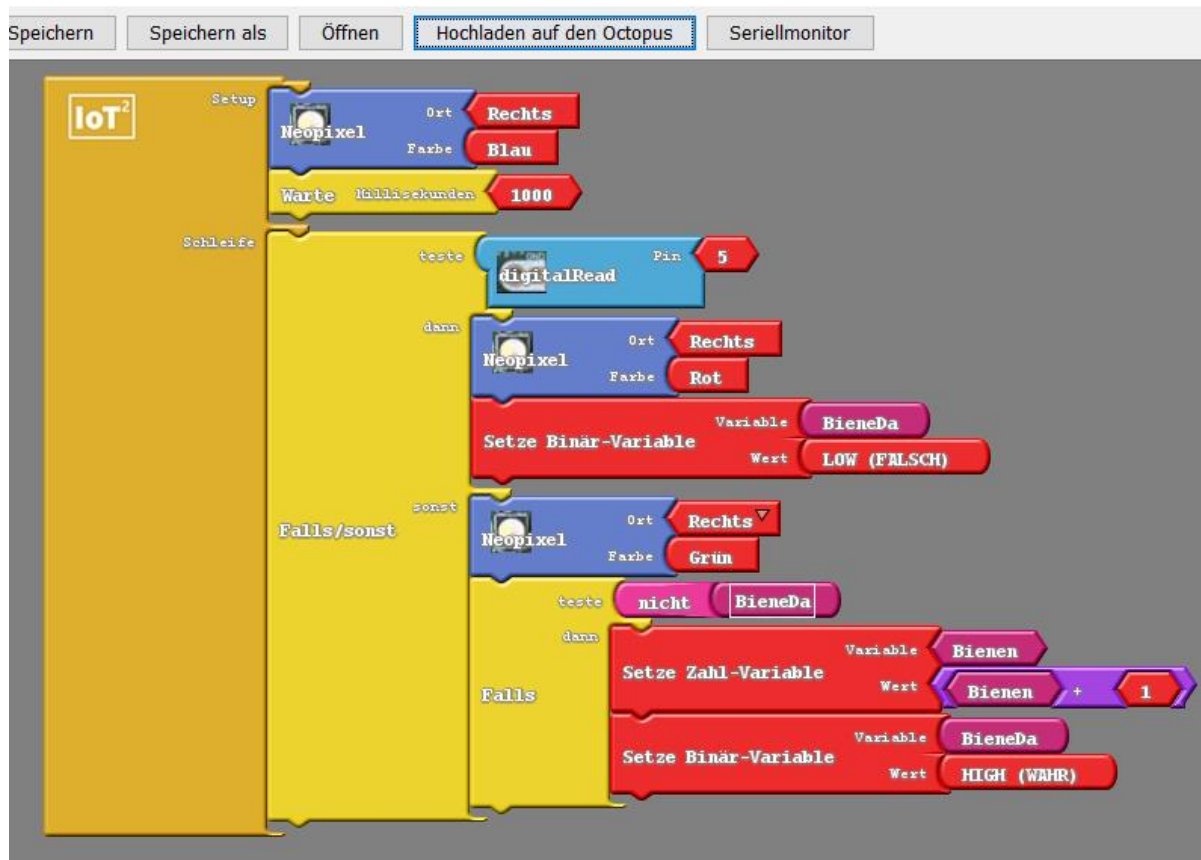
In den „dann“-Teil der „Falls“-Struktur habe ich die „Setze Binär-Variable“ aus dem „dann“ teil der „Falls/sonst“ Struktur übernommen. Den Wert änderte ich von „Low“ auf „High“. Dieser Baustein sorgt dafür, dass jede Biene nur einmal pro Durchflug gezählt wird und nicht in jedem Durchgang von dem Arduino durch die Schleife.



Nach Fertigstellung des Programms, muss es nur noch auf den Octopus hochgeladen werden.

Den Seriell Monitor (im Bild oben rechts) habe ich geöffnet. Über diesen kann man sehen, wie viele Bienen die Lichtschranke ausgelöst haben. Es erscheint sehr oft die gleiche Zahl,

da der Arduino das Programm in nur einer Millisekunde durchläuft, weshalb nicht die Anzahl der Zahlen entscheidend ist sondern die Höhe.



So sieht der Seriell Monitor dann bei 14 gezählten Bienen aus.

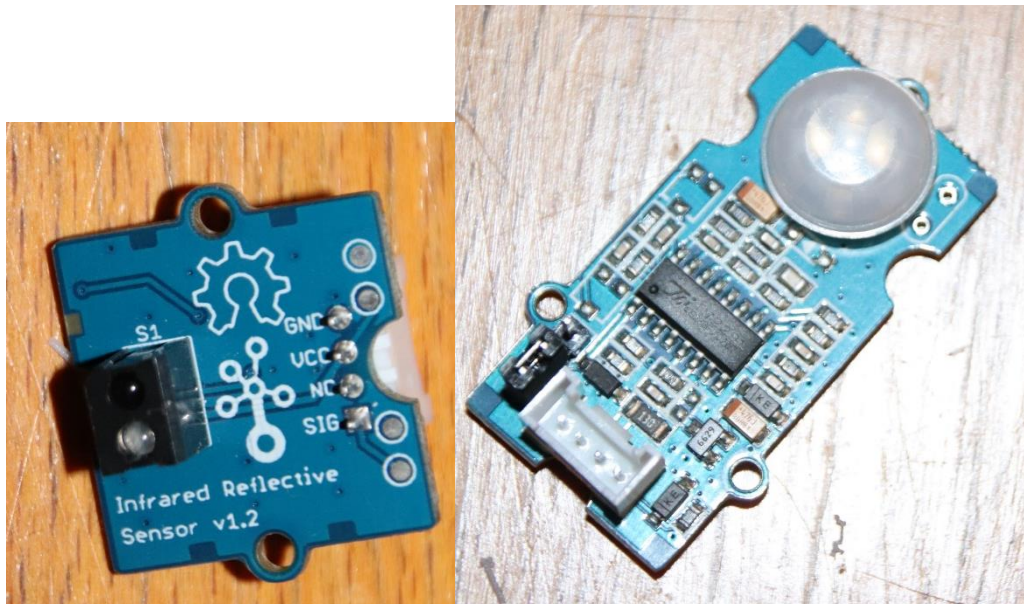
```
--  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14  
14
```

Auswahl des Sensors

Die beschriebene Lichtschranke war nicht der einzige Sensor, der für die Zählung in Frage kam. Es standen zunächst auch noch eine andere Lichtschranke sowie ein Bewegungsmelder für den Arduino zur Auswahl.

Die alternative Lichtschranke hatte jedoch einen zu großen Bereich zwischen dem Sensor und dem Zählpunkt. Das hätte bei einer Zählung zur Folge, dass Bienen, die sich lediglich um das Loch tummeln, ebenfalls erkannt und gezählt würden. Deshalb habe ich mich gegen diese Lichtschranke entschieden.

Der Bewegungsmelder benötigt 3,3 - 5 Volt. Da der Arduino Octupos jedoch nur 3 Volt liefert, konnte auch dieser nicht verwendet werden.



Links der letztendliche Sensor und rechts der Bewegungsmelder

Eigenschaften des Sensors

Auf dem oberen Bild ist gut erkennbar, dass am Stecker des Sensor hinten und vorne ein kleiner schwarzer Kasten sitzt, der zwei Linsen beinhaltet. Durch eine Linse wird ein Strahl ausgeworfen und durch die zweite Linse wird er empfangen.

Der Sensor benötigt eine möglichst helle Fläche auf der gegenüberliegenden Seite des schwarzen Kästchens, von der die Strahlen reflektiert werden können. Wird dieser Strahl unterbrochen, löst der Sensor aus und meldet das dem Computer. Daraufhin erhöht sich der Wert um eins. Der Sensor soll bei der späteren Zählung links vom Einflugloch der Bienen

montiert werden. Rechts soll eine möglichst gut geeignete Reflektionsfläche montiert werden.

Weitere Projektumsetzung

Das Projekt wird im Jahr 2022 weitergeführt. Das Messgerät wird, sobald die ersten Bienen fliegen, bei einem Imker vor ein Einflugloch eines Bienenstocks gehängt. Es überwacht die Bienen das ganze Jahr, um einen Jahresvergleich zu ermöglichen.

Für den Sensor selbst und den Arduino muss noch eine Schutzhülle entworfen werden und ggf. mit einem 3D-Drucker ausgedruckt werden.

Das Wetter wird entweder über eine in der Nähe befindliche Wetterstation dokumentiert oder über den Arduino selbst, denn das Octuposboard hat einen integrierten Wettersensor.

Unterstützungsleistungen

Herr Thomas Kübler von der VSK-Technik Kübler GmbH, Worms, hat mich bei der Programmierung des Arduino unterstützt.