

Der digitale Bienenstock

Bestäuber schützen und unterstützen

Von Jan Altenburg

Jugend forscht 2025

Gesamtschule Brüggen

Inhaltsverzeichnis

1. Deckblatt.....	1
2. Inhaltsverzeichnis.....	2
3. Einleitung.....	3
4. Motivation und Fragestellung.....	4
5. Verschiedene Ansätze ein Ziel.....	4
6. Der BienenHUB.....	5
7. Der BienenCounter, KI-basiert.....	8
8. Datenverarbeitung.....	10
9. Verwendung durch den Imker.....	13
10. Vergleich mit anderen Systemen.....	14
11. Ergebnisdiskussion.....	15
12. Fazit und Ausblick.....	15
13. Abbildungsverzeichnis.....	16
14. Literaturverzeichnis.....	16
15. Unterstützungsleistungen.....	17

Einleitung

Ein Imker muss seine Bienen regelmäßig überprüfen, um sicher zu stellen, dass sie gesund sind und dass sie sich der Jahreszeit entsprechend verhalten und entwickeln. D.h. Aktivität im Frühjahr – Bruttätigkeit bis in den Herbst, Ende der Bruttätigkeit, Einlagerung von Honig oder nach dem Schleudern und Einfüttern die Einlagerung von Futter. Dazu muss er den Bienenstock regelmäßig öffnen, um hinein sehen zu können. Dabei stört er die Bienen und löst auch Stress aus, welcher sich negativ auf die Gesundheit des Bienenvolkes auswirkt. Vor allem aber muss er dafür persönlich anwesend sein.

In ungewollten Situationen, wie z.B. der Bildung eines Schwarms oder einem Überfall durch andere Bienen, muss er zügig eingreifen, um Verluste von Bienen und Honig zu vermeiden. Das kann er aber nur, wenn er es auch mitbekommt.

Mein Projekt beschäftigt sich mit der Möglichkeit für den Imker, seine Bienenvölker aus der Ferne zu beobachten und zu überwachen, um so die Eingriffe in den Bienenstock seltener und gezielter durchführen zu können. Durch technische Überwachungsmöglichkeiten wird ein Einblick in/um das Bienenvolk ermöglicht, ohne den Stock öffnen zu müssen. Ferner gibt es eine Warnung, wenn eine kritische Situation eintritt. Meine Systeme bieten eine höhere Anzahl an digital überwachbaren Bienenstöcken als andere Systeme, außerdem viel genauere Warnungen über Schwarmtätigkeit oder Räuber-Aktivitäten.

Die dafür notwendige Gewichts- und Temperaturüberwachung des Bienenstocks wird über den BienenHUB¹ realisiert. Das Gewicht gibt Aufschluss über die Menge des eingelagerten Honigs bzw. Futters. Die Brutraumtemperatur gibt Aufschluss über den Zustand des Bienenvolkes (Bruttätigkeit + Zustand/Gesundheit von der Königin und dem Bienenvolk). Ein normales Bienenvolk hat im Sommer eine ungefähre Brutraumtemperatur von 34 °C. Eine stark schwankende Temperatur kann auf ein Problem² mit der Königin hinweisen. Ein Abfall der Temperatur zeigt u.U. das Ende der Bruttätigkeit im Herbst und die Bildung einer Wintertraube an und erleichtert die Planung der Winter-Milben-Behandlung.

Mittels des von mir entwickelten KI-basierten BienenCounters ist es möglich, Bienen auf einer optischen Ebene zu erkennen, zu verfolgen und zu zählen. So sind so relativ genaue bzw. zuverlässige Alarme über Bienenschwärme oder Räuberbienen möglich.

Im Vergleich zu käuflich erwerbbaaren oder Selbstbauversionen aus dem Internet, ist mein System deutlich günstiger im Bau, einfacher in der Handhabung und erweiterbar.

Um das Projekt realisieren zu können habe ich mich mit dem Entwickeln und Löten von Platinen, dem Programmieren von Mikrokontrollern unter Einbindung von verschiedensten Komponenten unterschiedlicher Herkunft, dem 3D-Druck, verschiedenen Datenübertragungstechniken, Datenverarbeitung und Visualisierung sowie einem KI-Model zum Erkennen und Tracken von Objekten beschäftigt.

¹ Der BienenHUB ist mein selbst entworfenes Produkt, welches das Gewicht sowie die Temperatur der Bienenstöcke ermittelt und übermittelt

² Krankheit, Alter oder Tod

Motivation und Fragestellung

Meine Bienenvölker befinden sich nicht bei mir zu Hause. Da ich mich aber noch am Anfang der Imkerei befinde ist es schwer, Sachen nach Gefühl zu machen. So hilft das Beobachten bzw. Messen. Da ich mich ebenfalls generell für Technik interessiere, bin ich schnell auf das Thema Stockwaagen gekommen. Da die käuflich erwerbbaaren oder Selbstbauwaagen für meine Bedürfnisse und Geldbeutel nicht passend waren, habe ich angefangen mein eigenes System zu entwickeln. Hierzu habe ich den Bauteilmarkt nach geeigneten und günstigen Einzelkomponenten abgesucht, diese Hardware technisch kombiniert und so programmiert, dass eine funktionsfähige Einheit entstanden ist. Ich habe mich zuerst an zwei Selbstbaumodellen probiert, denn kaufbare Systeme sind sehr teuer. Das Erste funktionierte nicht richtig. Das Zweite war sehr kompliziert zu bauen und für mich (damals 14 Jahre) nicht nachzuvollziehen. Hier konnte man zwar noch einen Bienenzähler mit anbauen, dieser beschränkte sich aber auf einfache Lichtschranken, die die Bienen zählen. Auch der Aufbau war kompliziert. Man musste zum Beispiel SMD-Löten können und die Teile waren schwierig zu beschaffen. So kam es dazu, dass ich mich selbst mit dem Entwurf einer Stockwaage und später einem Bienenzähler beschäftigte. Ich wollte Systeme schaffen, die mit einfachen Mitteln herstellbar sind, keine besonderen handwerklichen Fähigkeiten benötigen und gut finanzierbar sind.

Verschiedene Ansätze ein Ziel

Für mein Endprodukt habe ich viele verschiedene Ansätze verfolgt, um das für mich optimale System zu bauen. Ich habe mit der Hardware sowie der Software umfangreich variiert und auch verschiedenste Datenübertragungsmöglichkeiten getestet. Das erste komplette System war der BienenHUB in der Version 1.0.1. Dieses System unterstützte 3 Waagen zusammen mit 3 Temperatursensoren. Mittlerweile lassen sich auch noch mehr Waagen (für mehr Bienenvölker) anschließen. Die Software in der Arduino IDE geschrieben, lief auf

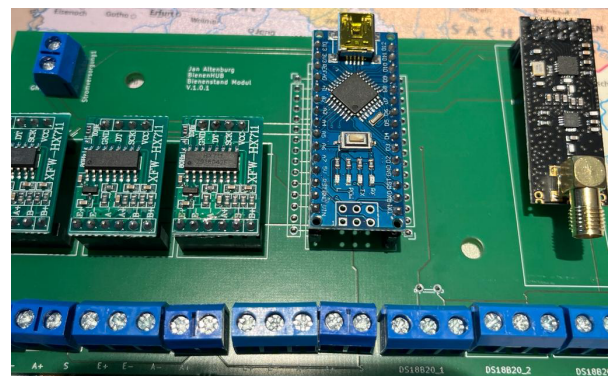


Abbildung 1 Platine BienenHUB V.1.0.1 (Foto: Jan

Altenburg)

einem Arduino Nano. Die Übertragung der Daten fand über ein NRF-Modul statt. Die Daten wurden mittels des NRF24L01+ Modul an eine zweite Station übermittelt (Abb. 2 (1)), die diese dann über WLAN in meine Datenbank speicherte. In der Anwendung am Bienenstand traten dann Probleme bei der Funkverbindung zwischen den beiden NRF-Modulen auf. Ich habe es mit handelsüblichen 433 MHz-Modul Funkstrecken versucht, aber gleiche negative Erfahrungen gemacht. Ich wollte keine normale WLAN-Verbindung benutzen, da das Produkt auch an etwas abgelegenen Orten funktionieren sollte. Deshalb das Prinzip mit der

Funkstrecke. Ich habe ebenfalls die CPU's (Kontrolleinheiten) variiert, um verschiedene Funktionen sowie die Anzahl der verfügbaren Anschlüsse testen zu können.

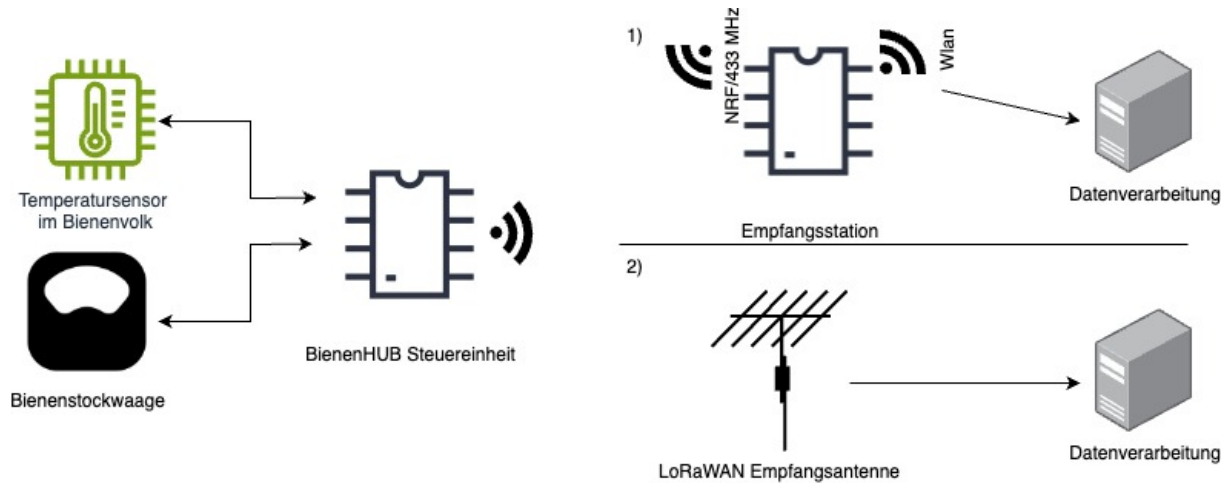


Abbildung 2 Verschiedene Ansätze für den BienenHUB (erstellt mit draw.io)

Der BienenHUB

Letztendlich ist die Wahl auf das Heltec WiFi Lora v3 Modul gefallen, als zentrale Steuereinheit. Diese ESP32 basierte CPU mit eingebautem Lora Modul bietet viele Vorteile für mein Projekt. Besonders wichtig sind die gute Leistung des ESP32 und das integrierte Lora Modul. Denn ich benutze jetzt für den BienenHUB das LoraWAN³ Netzwerk zum Übertragen meiner Daten (Abb. 2 (2)). Dieses Netzwerk ist ein öffentliches und für die Landwirtschaft ausgebautes System, um Daten unabhängig von Mobilfunk oder WLAN übertragen zu können.

Der BienenHUB ermittelt die Gewichtsdaten mittels eines HX711 und einer Bosche H30A Wägezelle. Der HX711 ist ein 24 Bit A/D Wandler, der die Signale der Wägezelle umwandelt und an die CPU weitergibt. Pro HX711 können theoretisch zwei Wägezellen angeschlossen werden. Da aber die Auflösung (=Genauigkeit der Waage) für die zweite Wägezelle niedriger ist, verwende ich nur eine Waage pro HX711. Bei der H30A von Bosche handelt es sich um eine Genauigkeitsklasse C3 Wägezelle. „Die Genauigkeitsklasse C3 bedeutet zum Beispiel, dass die Zelle in Wägesysteme eingesetzt werden kann, die Gewichtsveränderungen bis zu 1/3.000 der maximalen Nennlast des Systems messen können.“ [2] Da ich Wägezellen mit einer Nennlast von 150 kg verwende liegt die Genauigkeit also bei 50g.

³ LoraWAN (Long Range Wide Area Network) ermöglicht energieeffiziente Datenübertragung über große Entfernungen [1]

Ich habe beim Aufbau Tests mit Beutenbestandteilen durchgeführt, die eine Genauigkeit von 10g gezeigt haben. Allerdings nur im unteren 1/6 des Messbereiches. Ich konnte aber im Hauptanwendungsbereich eine gute Genauigkeit verifizieren.

Um die Temperatur im Brutraum des Bienenstockes messen zu können (Bruttätigkeit + Zustand/Gesundheit des Bienenvolkes) kommt ein DS18B20 Temperatursensor zum Einsatz. Dieser ist klein und wasserdicht, eignet sich also perfekt für die engen Wabengassen in der Bienenbeute. Der DS18B20 wird über den One-Wire-Bus angesteuert. In diesem System können mehrere Sensoren über eine DATA-Verbindung verwendet werden. Das spart GPIOs⁴ an der CPU. So können noch andere Sensoren angeschlossen werden, zum Beispiel noch mehr HX711 Module. Der DS18B20 Sensor hat einen Messbereich von -55°C bis 125°C bei einer Genauigkeit von +/-0,5°C.

Die Stromversorgung habe ich anfänglich einfach mittels USB C Kabel realisiert. Um aber unabhängig sein zu können, verwende ich das LoraWAN Netzwerk zusammen mit einem selbstgebauten Solar-Akku-System. Das 12V Solarpanel ist mit einem von mir selbstentwickelten Haltesystem montiert und verbunden mit einem BMS 20A Laderegler. Dieser lädt 2x 26650 Li-Ion Akkus, die in Reihe geschaltet sind. So bietet das System eine Spannung von 7,2V, um auch die Waagen mittels Spannungswandler mit 5V versorgen zu können. Ich habe zuerst mit einem 9V System experimentiert, ein 9V Solarpanel mit 2x 18650⁵ Akkus. Dieses System war allerdings nicht leistungsstark genug. Die Kapazität war deutlich niedriger und die 9V Solarpaneele lieferten beim Laden nur 3W. Das neue Panel mit 12V liefert bis zu 20W.



Abbildung 3 Solarpanel mit Haltesystem (Foto: Jan Altenburg)

Um die verschiedenen Komponenten miteinander verbinden zu können benötige ich Leiterplatten. Diese habe ich zuerst selbst hergestellt. Aus Zeit- und auch Qualitätsgründen habe ich dann die Platinen in KiCad⁶ (Abb. 5) erstellt, industriell bei PCBWay und Aisler.net fertigen lassen und später mit einzelnen Bauteilen bestückt. Das hat den Zeitaufwand deutlich reduziert, um ein System zusammen zu bauen und vorzubereiten.

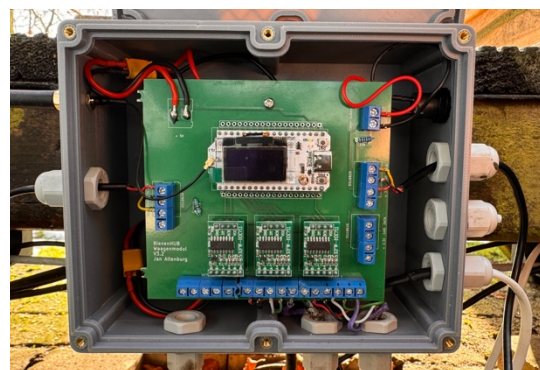


Abbildung 4 Bestückte Platine BienenHUB V3.2 im Gehäuse; Endversion (Foto: Jan Altenburg)

⁴ Anschlusspins an dem Microcontroller

⁵ 18650 Akkus sind sehr bekannt/beliebt im Bastler Bereich

⁶ KiCad = Kostenfreies PCB (Leiterplatten) Erstellprogramm

Um die bestückten Platinen dann am Bienenstand wassergeschützt anbringen zu können, fertige ich meine individuellen Gehäuse mittels eines 3D-Druckers. Ich verwende das Autodesk Programm Fusion 360. In Fusion konstruiere ich wasserdichte und funktionelle Gehäuse inklusive Deckel. In Abbildung 6 ist ein Schnitt durch ein Gehäuse (rosa) inkl. Deckels (gelb) dargestellt.

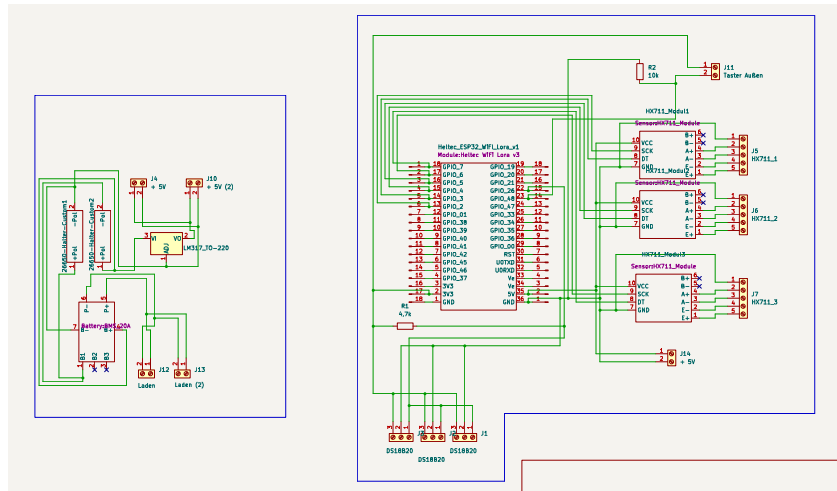


Abbildung 5 Schaltplan der Platine (Screenshot aus KiCad)

Der Deckel wird durch Senkkopfgewindeschrauben mit dem Gehäuse verbunden. Im Gehäuse werden dazu Einschmelzhülsen eingeschmolzen. Dazu dienen die Röhren im Gehäuse (rosa). Durch ein Nut- und Federsystem schließt der Deckel wasserdicht ab. Zur Abdichtung gieße ich immer eine dünne Schicht Silikon in die Nut. Wenn dann der Deckel durch die Schrauben angepresst wird, schließt der Deckel noch besser. Durch die Verwendung von Senkkopfschrauben wird auch die Röhre, die der Befestigung dient abgedichtet, um zu verhindern, dass sich Wasser irgendwo sammeln kann. Beim 3D Drucken des Gehäuses ist auf das Material zu achten. Ich verwende dazu ASA. Das Acrylnitril-Styrol-Acrylat, welches ich von der Firma Bambulab beziehe, bietet eine sehr gute UV-Beständigkeit sowie eine Temperaturresistenz bis 100 °C. So ist es perfekt für den Einsatz im Sommer im Außenbereich geeignet.

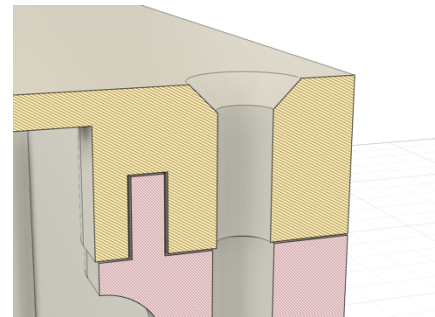


Abbildung 6 Deckel und Gehäuse

Verbindung (Screenshot aus Fusion 360)

Die Datenübertragung vom BienenHUB findet mittels LoraWAN statt. Dafür müssen die Daten mittels Payload formatter⁷ in eine ganze Zahl umgewandelt werden, da sie nur so übertragen werden können. Die Daten werden dann über die BienenHUB-Antenne an die nächste LoraWAN Empfangsantenne übertragen. Diese befindet sich in meinem Fall auf dem örtlichen Rathaus. Von dort werden die Daten an den Server eu1 in Irland übertragen. Dort werden die Daten dann zurück in die Ausgangsform umgewandelt und können weiterverarbeitet werden.

Um die Zusammenarbeit der einzelnen Komponenten mit dem Arduino und die Datenübertragung realisieren zu können, musste ich ein Programm schreiben, welches das ermöglicht. Nach zahlreichen Anpassungen und Korrekturen läuft das System zuverlässig und störungsfrei.

⁷ Payload formatter = Teil des Programmcodes um Messwerte umzuwandeln

Der BienenCounter

Der BienenCounter ist ein von mir entwickeltes KI basiertes Programm, das graphisch Bienen erkennen, verfolgen und zählen kann.

Der BienenCounter bezieht sein Videosignal von einer IP-Überwachungskamera mittels RTSP-Stream⁸. Ich habe nach verschiedenen Versuchen einen Holzkastenvorbau für den Bienenstock konstruiert (Abb. 7). Dieser gibt den Bienen den Weg in den Bienenstock vor. Das erleichtert die Erkennung der Bienen, da immer ähnliche Bilder geliefert werden müssen. Dies wird in der Verarbeitung des Videos später wichtig. Durch eine Luftzufuhr wird im Sommer der Kasten belüftet, um es den Bienen angenehmer zu gestalten. Außerdem kann so Feuchtigkeit, die beim Trocknen des Honigs entsteht, besser verdunsten. Oben auf dem Kasten befindet sich die Kameraeinheit, die durch den Deckel in den Innenraum filmen kann (siehe Abb. 8). Ich habe mich bei der Kamera für eine kompakte Hikvision Kamera entschieden, genauer für die DS-2CD2085FWD-I. Diese Kamera ist klein und wird mit POE⁹ angeschlossen. Dies macht es einfach, sie in mein bestehendes UniFi Netzwerk zu integrieren und einzurichten. Ich habe mich bei der Marke explizit für Hikvision entschieden. Ich hatte u.a. Reolink und Raspberry Pi Kameras im Einsatz, dort aber Probleme mit dem RTSP-Stream, der ausreichenden Stromversorgung und Kompatibilität. Bei Hikvision handelt es sich um einen namenhaften und in der Industrie oft verwendeten Hersteller, der

eine gute Qualität liefert. Damit konnte ich einen stabilen Stream für eine zuverlässige und genaue Erkennung erreichen. Das Videosignal wird auf einem lokalen Computer verarbeitet. Wichtig ist eine große Rechenleistung des Computers, da KI-Applikationen große Rechenkapazitäten benötigen. In einem von mir entwickelten Python Programm werden mittels der Yolo v8 KI und einem custom Model¹⁰ die Bienen erkannt. Ich habe mit selbst erstelltem umfangreichem Bildmaterial meiner Bienen ein Yolo v8 Model¹¹ trainiert. So ist das Modell genau auf meine Anwendung und Umgebung angepasst. Mittels der „linecounter“ Funktion aus der Supervision Bibliothek ist es möglich, die getrackten Bienen an einer definierten Linie zu zählen. Linecounter kann unterscheiden in welche Richtung sich die getrackten Objekte bewegen. So ist die



Abbildung 7 Vorbau am Bienenstock

(Foto: Jan Altenburg)

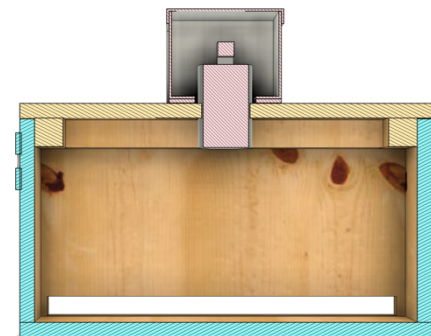


Abbildung 8 Konstruktion des Vorbaus

(Screenshot aus Fusion 360)

⁸ Ein RTSP-Stream ist ein Protokoll, um Videosignale über IP zu übertragen.

⁹ Power-Over-Ethernet: Die Stromversorgung wird über die Ethernet Datenverbindung realisiert

¹⁰ Modell: Informationen über die erkennbaren Objekte

¹¹ Yolo v8 = KI-Modell der bekannten Yolo KI (Version 8) von Roboflow

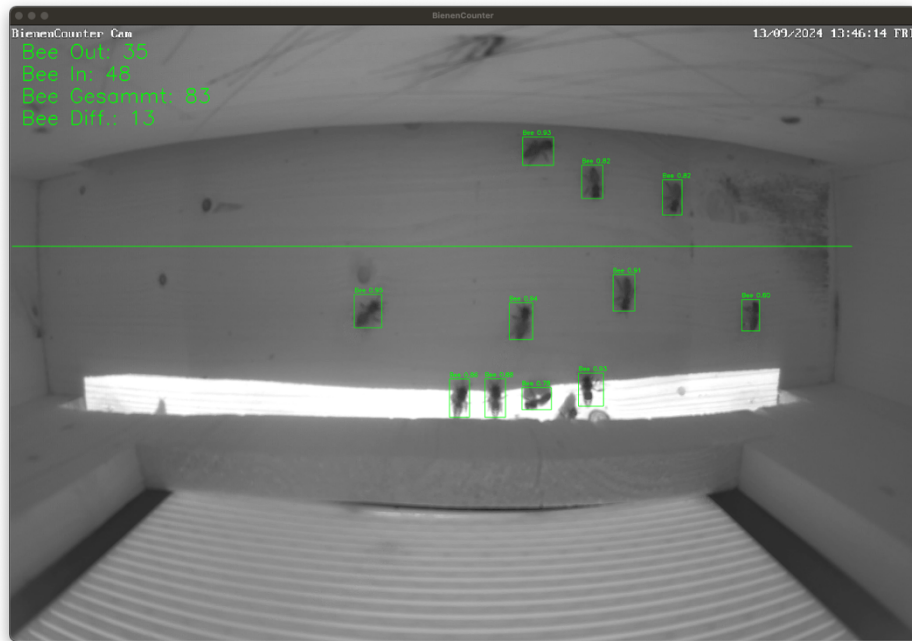


Abbildung 9 BienenCounter Visuelle Ausgabe (Screenshot aus der Visuellen Ausgabe von PyCharm)

Angabe von in bzw. out Werten möglich. Die Differenz zwischen den beiden Werten gibt über längere Zeit Auskunft darüber, ob Bienen verloren gegangen sind oder ob Bienen dazu gekommen sind.

Ein sehr schnell sinkender Differenzwert zeigt einen Bienenschwarm an, ein sprunghaft steigender zeigt Räuberbienen an.

Um sicher zu stellen, dass die Bienenzählung korrekt ist, habe ich mein Messgerät überprüft. Dazu habe ich kurze Videos zu verschiedenen Uhrzeiten mit unterschiedlichen Lichtverhältnissen und variierender Bienenichte verwendet. Diese habe ich auf meinem Hauptrechner durch die KI auswerten lassen und parallel manuell die Bienenbewegungen mitgezählt.

Der Vorteil am BienenCounter, im Vergleich zu den Stockwaagen, liegt darin, dass die Aussagen über Räuberbienen und Schwärme deutlich schneller und genauer getätigt werden können. Da Stockwaagen den Schwarmalarm ausschließlich an einem Gewichtsverlust festmachen, kann dieser schnell fehlerhaft sein. Außerdem senden Stockwaagen normalerweise immer in bestimmten Zeitintervallen von 10, 15 oder 20 Minuten. So kann es eventuell lange dauern, bis es zu einer Meldung eines Ereignisses kommt. Der BienenCounter wertet die Daten fast in Echtzeit aus und sendet einmal pro Minute die Ergebnisse. So kann beispielsweise ein Bienenschwarm innerhalb von wenigen Minuten voll automatisch erkannt werden. Der Imker wird benachrichtigt und kann sofort eingreifen, um den Bienenschwarm einzufangen. Das ist wichtig, denn die meisten Bienenschwärme überleben in freier Wildbahn nicht lange. Dies liegt meistens an Milben, die in Bienenvölkern vorkommen und normalerweise durch den Imker bekämpft werden. Außerdem bedeutet ein Schwarm auch immer einen Mitnahmeverlust an Honig.

Die ermittelten Informationen werden durch das MQTT-Protokoll¹² für die weitere Verarbeitung übertragen.

¹² MQTT = Netzwerkprotokoll für den Datenaustausch

Datenverarbeitung

Eine zuverlässig funktionierende Datenverarbeitung ist ein wichtiger und einer der zeitaufwändigsten Bestandteile meines Projektes. Es ist wichtig, dass die Daten korrekt behandelt und gedeutet werden. Meine Daten werden gesammelt, gespeichert und visualisiert. Darüber hinaus ist es meinen Systemen möglich, mit diesen Daten weiterzuarbeiten, was ich sonst von keinem anderen System kenne. So kann beispielsweise der Honiganteil im Bienenvolk berechnet werden; im Winter der Futtermenge.

Allerdings müssen die Daten dazu erstmal ins System kommen.

Der BienenHUB 3.2 überträgt seine Daten mittels LoraWAN in das TTN-Netzwerk auf Server eu1. Aufgrund von mir zugewiesener Keys sind meine Daten erkennbar und werden ausschließlich für mich sichtbar gemacht. Die Daten sind verschlüsselt und können nur mit den passenden Keys empfangen werden. Vom TTN-Server kann ich anschließend mittels MQTT die Daten abrufen und in NodeRED¹³ weiterverarbeiten. NodeRED schreibt die Daten anschließend in eine InfluxDB¹⁴, wo sie für die Zukunft sicher gespeichert werden. Außerdem schreibt NodeRED die Daten auf meinen lokalen MQTT-Broker.

So kann Home Assistant¹⁵ die Daten abrufen und weiterverarbeiten. Ich habe Automatisierungen erstellt, die den Honiganteil im Bienenvolk berechnen. Der Imker gibt immer noch eine Arbeit im Bienenvolk Informationen in eine Eingabemaske ein. Diese beziehen sich aufs

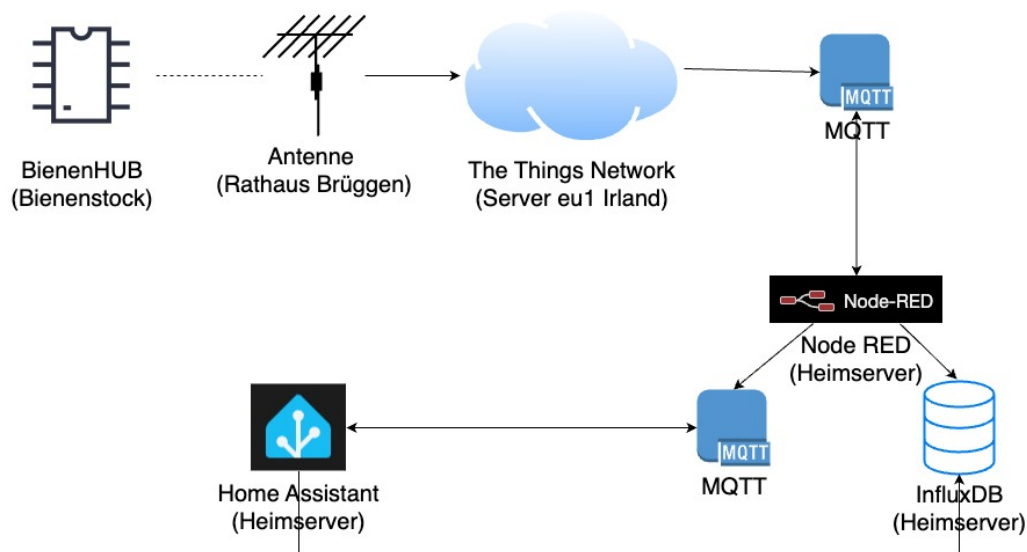


Abbildung 10 Ablauf der Datenverarbeitung beim BienenHUB (Erstellt mit draw.io)

¹³ NodeRED = Smarthome Software basieren auf graphischen Nodes

¹⁴ Eine InfluxDB ist eine Time-Series Database, also eine Datenbank, die ihre Informationen mit Zeit Informationen speichert.

¹⁵ Home Assistant ist ein kostenloses Open Source Projekt von der OpenHomeFoundation für Heimautomatisierung (welches recht bekannt ist)

Bienenvolk selbst und das verwendete Material (z.B. Aufsetzten von Honigräumen oder Bruträhmchen; führt zu Gewichtsveränderungen). Wenn sich nichts verändert, muss auch nichts eingetragen werden. Der Futterwert¹⁶ wird zusammen mit den ermittelten Daten in Graphen sowie Anzeigen visualisiert und angezeigt. Der ermittelte Futterwert wird auch mittels MQTT und NodeRED in die InfluxDB geschrieben. Der Imker kann auch mittels Grafana¹⁷, welches lokal auf Home Assistant installiert ist, und den InfluxDB Daten sich Langzeit-Daten anschauen. So kann theoretisch bis zum Anfang der Aufzeichnungen zurückgeschaut werden. Der aktuelle Entwicklungsstand kann mit dem vor z.B. einem Jahr verglichen werden.

Der BienenCounter überträgt seine Informationen mittels MQTT auf das lokale NodeRED. Da die Bienen sich nicht bei mir zu Hause befinden, muss ich einen Zwischenschritt gehen, um die Daten auf meinen Heimserver zu bringen. Das lokale NodeRED bei den Bienen ist in der Lage die Daten in meine InfluxDB zu schreiben. Wenn sich dort etwas ändert, stellt das meine Heimserver NodeRED Applikation fest und schreibt die neuen Daten auf den MQTT-Broker.

So kann Home Assistant diese wieder abgreifen und weiterverarbeiten.

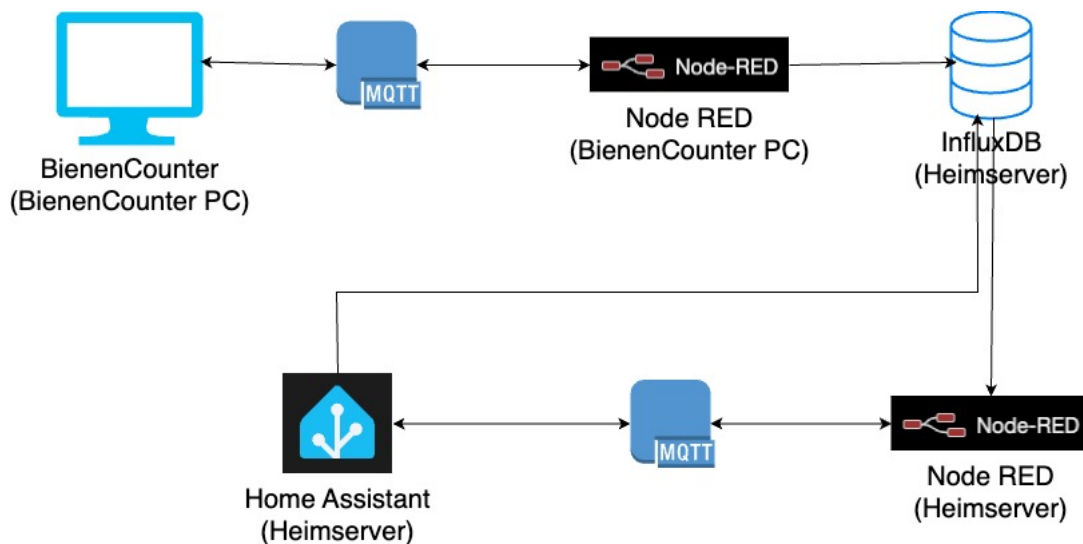


Abbildung 11 Ablauf der Datenverarbeitung beim BienenCounter (Erstellt mit draw.io)

In Home Assistant werden die Daten vom BienenCounter ebenfalls visualisiert, wie beim BienenHUB. Mit Automatisierungen werden Schwärme und Räuberbienen erkannt. Die Automatisierung wird automatisch aktiviert, wenn der Bienen-Differenzwert unter oder über einen bestimmten Grenzwert sinkt bzw. steigt. Diese Automatisierung arbeitet aber nur am Tag, zwischen Sonnenaufgang und Sonnenuntergang. Denn es kommt besonders im Sommer vor, wenn die Bienen den Honig trocknen, dass sich Bienen am Flugloch befinden und mit ihren Flügeln fächern. Das machen sie zur Temperierung und zur Entfeuchtung des Bienenstockes. Dann ist unter Umständen der ganze Bereich vor dem Bienenstock voll mit Bienen und es kann zu Fehlerkennungen kommen. So verhindere ich Fehlalarme in der Nacht. Wenn es Tag ist wird unterschieden, ob ein Schwarm oder Räuberbienen vorliegen.

¹⁶ Der Futterwert gibt an wie viel Honig/Futter sich im Bienenvolk befindet

¹⁷ Grafana ist, auch in der Industrie angewendete, Visualisierungssoftware, welche mittels Datenbanken Daten in Graphen oder anderen Feldern anzeigen kann

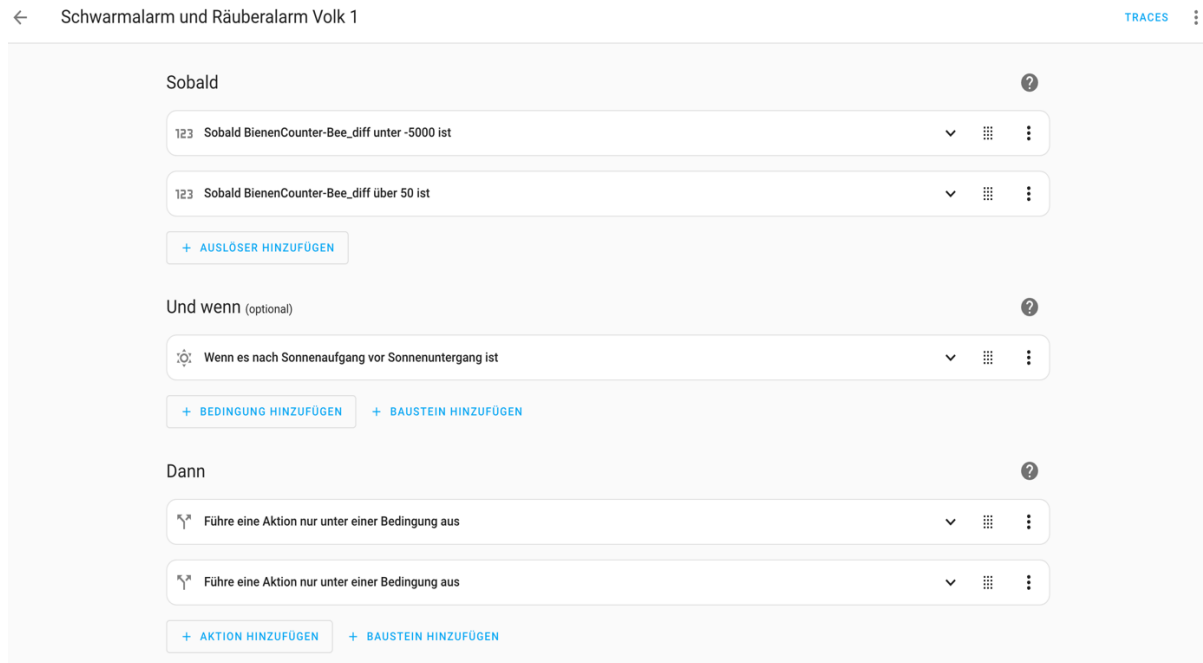


Abbildung 12 Automatisierung zur Schwarm- und Räubererkennung (Screenshot aus der Home Assistant Web GUI)

Der Grenzwert für einen Schwarm liegt bei einer Differenz von -5000 Bienen. Dieser Wert entspricht ungefähr einem kleinen Bienenschwarm. Räuberalarm wird bei einer Differenz von +50 Bienen ausgelöst. Der Schwarmwert ist im Vergleich zum Räuberwert sehr hoch angesiedelt. Das liegt daran, dass der Differenzwert immer im negativen Bereich liegt. Denn Bienen sterben draußen und gerade am Morgen verlassen die Bienen erstmal den Bienenstock, bevor sie wieder reinkommen. So verhindere ich Fehlalarme. Dass neue Bienen hinzukommen, sollte eigentlich nicht vorkommen. Deshalb ist dieser Wert so niedrig. Sollte mal ein Schwarm oder eine Räuberaktion erkannt werden, wird alle 10 Sekunden eine Mitteilung ans Handy oder Tablet gesendet. Dies kann durch eine Bestätigung in der App beendet werden. Das häufige Senden der Mitteilung erhöht die Wahrscheinlichkeit, dass man es mitbekommt und darauf reagieren kann.

Verwendung durch den Imker

Diese ganzen Erkenntnisse müssen nun übersichtlich und einfach zugänglich für den Imker bereitgestellt werden. Sonst bringen die Informationen nichts.

Dazu verwende ich die Handy-App von Home Assistant. Hier lege ich ein Dashboard mit mehreren Unterkacheln an, die mir alle wichtigen Informationen bereitstellen, ohne dass man lange suchen muss. Die erste Seite zeigt die wichtigsten Informationen an (Abb. 13). Hier werden die Messwerte sowie die berechneten Werte angezeigt und schnell verfügbar gemacht. Außerdem kann hier ein Schwarm oder Räuberalarm bestätigt werden, um den Alarm zu beenden. Ein schneller Blick zeigt, ob es den Bienen gut geht. Einen Tab weiter (Abb. 14) werden die Informationen über erfolgte Aktionen am Stock bezüglich des Materials eingegeben. Hier kann die Tabelle schnell nach einem Eingriff bearbeitet werden. Der dritte Tab (Abb. 15) zeigt Informationen über den BienenCounter. Hier können Messwerte wie das Gewicht des Bienenstockes, der Futter/Honig Anteil, die Brutraumtemperatur und der Ertrag der letzten 24 Stunden abgelesen werden.

Die Dashboards können über die App oder auch mittels eines Webbrowsers eingesehen werden. So besteht immer die Möglichkeit, schnell einen Blick auf seine Bienen zu werfen.

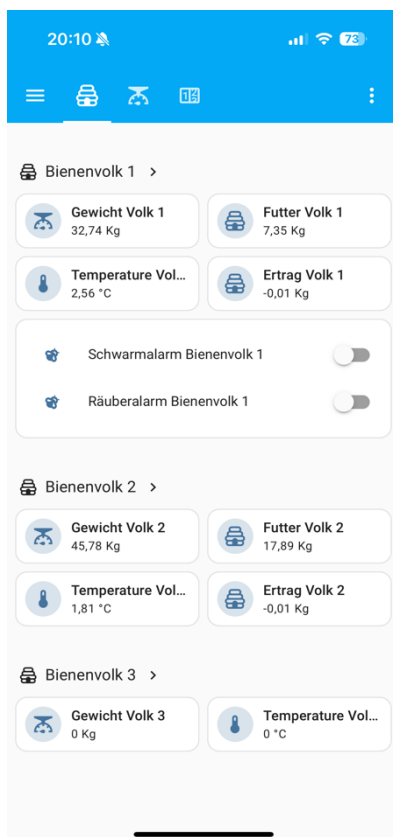


Abbildung 13 BienenDashboard1
(Screenshot aus Handy App)

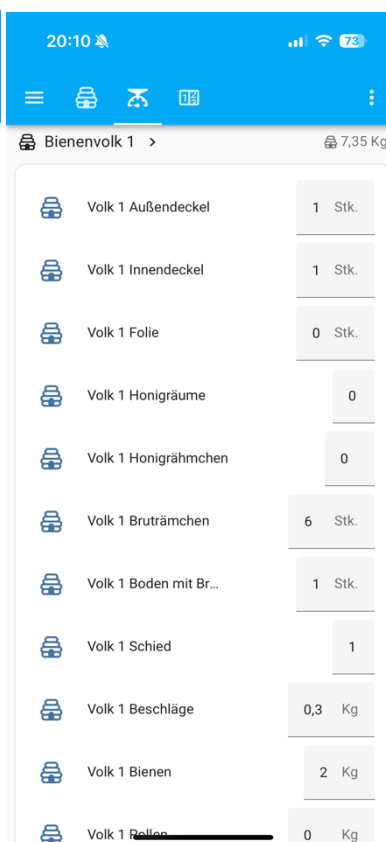


Abbildung 14 BienenDashboard2
(Screenshot aus Handy App)

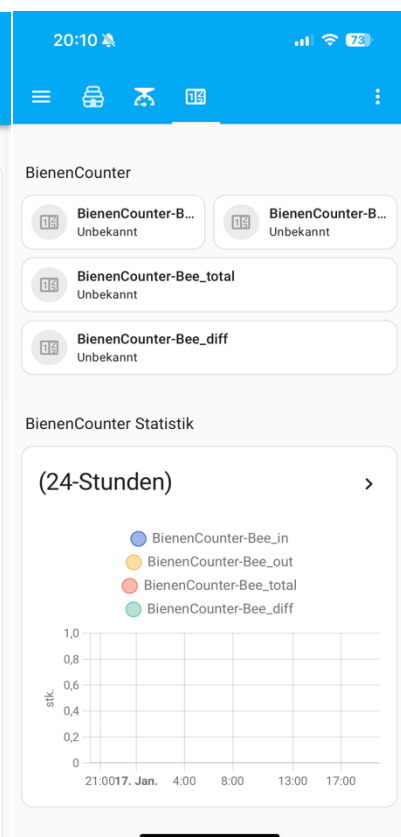


Abbildung 15 BienenDashboard3
(Screenshot aus Handy App)

Vergleich mit anderen Systemen

Es gibt mittlerweile einige Stockwaagen im Internet zu kaufen und auch Anleitungen zum Selbstbau. Doch mir hat immer irgendetwas gefehlt oder nicht gefallen, was sich schließlich zu diesem Projekt entwickelt hat. Ich vergleiche mein System mit dem Beelogger Universal in der Version 1.13a und dem ApiGraph 4 von Wolf Waagen. Den Beelogger Universal habe ich zum Teil gebaut und dabei getestet. Der ApiGraph 4 ist ein komplett fertig kaufbares System, das einen ähnlichen Funktionsumfang wie mein System hat. Diese Auswahl soll ein Stück weit den verfügbaren Markt widerspiegeln. Die Wolf Waage konnte ich nicht zum Testen kaufen, ich beziehe mich im Vergleich also auf die Aussagen von der Webseite des Herstellers. Ich verwende für diesen Vergleich die Wolf Waagen, da ich diese in YouTube schon öfter gesehen habe und deshalb ein gutes Verständnis von dem System habe.

Zuerst möchte ich über die Kosten sprechen, auch wenn dieser Punkt schwer zu vergleichen ist. Rein vom Material kann man sagen, dass sich die Selbstbauvariante Beelogger und der BienenHUB im selben Rahmen bewegen. Mein System benötigt allerdings deutlich weniger Teile. Ich komme für meinen Funktionsumfang mit weniger zurecht. Das macht den Einkauf der Teile deutlich einfacher. Mir ist beim Bau des Beeloggers aufgefallen, dass die Teilebeschaffung zum Teil kompliziert ist. Es werden sehr spezifische Teile benötigt, die dann schwer zu beschaffen waren. Die Kosten liegen im Bereich um 100€. Der ApiGraph 4 bewegt sich bei einer ähnlichen Konfiguration um die 1000€¹⁸ [3]. Zusätzlich benötigt man beim ApiGraph Abos für eine SIM-Karte und auch für die Onlinedatenverarbeitung.

Der Beelogger kann auch mit GSM verwendet werden, bietet aber auch andere Optionen. Preislich sind die Selbstbausysteme also sehr interessant.

Die Unterschiede der Selbstbausysteme zeigen sich dann beim Zusammenbau. Ich hatte bereits die Komplexität des Beeloggers beschrieben. Für die essenziellen Funktionen benötige ich mit meinem System weniger Zubehör, so ist der Bau meiner Waage auch deutlich einfacher.

Dies zeigt, dass ein auf die notwendigen Funktionen heruntergebrochenes System, ohne laufende Kosten für SIM-Karten, nicht allzu teuer und relativ einfach zu konzipieren ist.

Noch ein Wort zum BienenCounter. Es gibt ähnliche Systeme, die mit Lichtschranken arbeiten. Damit sie funktionieren, müssen die Bienen durch kleine Tunnel, wo nur eine Biene durchpasst. So behindere ich die Bienen bei ihrem normalen Verhalten, Sorge für einen Stau vor den Durchgängen in den Stoßzeiten und damit zu Stress. Außerdem können die Lichtschranken bei Kontakt mit den Bienen z.B. durch Pollen verschmutzen, und dann fehlerhaft messen.

Solche Probleme habe ich mit der Überwachungskamera und dem KI-Modell nicht. Die Kameralinse verschmutzt nicht so schnell und durch meinen Holzkasten behindere ich die Bienen nicht. Ich habe extra Holz als Naturstoff und angepasst an das Material der Beute gewählt, um wenig Irritation bei den Bienen auszulösen.

¹⁸ Konfiguration: ApiGraph 4, Dadant Unterbau, Brutraumtemperatursensor, Solar, SIM, Onlineverwaltung

Ergebnisdiskussion

Ich habe festgestellt wie kompliziert es ist, ein funktionierendes System auf die Beine zu stellen. Aber ich habe auch Wege gefunden, es möglich zu machen. Der BienenHUB erfüllt seine Aufgaben zuverlässig und komplett und auch die Visualisierung für den Imker am Ende funktioniert sehr gut. Die Anwendung und eventuelle Erweiterungen sind sehr einfach durchzuführen.

Der BienenHUB stellte mich vor besondere Herausforderungen. Das Programmieren einer Steuereinheit, die so unterschiedliche Komponenten miteinander verbindet ist sehr zeitaufwändig. Es bedarf vieler Anpassungen und Korrekturen. Seitdem die Verbindungen stehen, funktioniert er problemlos.

Der programmtechnische Teil vom BienenCounter funktionierte in der Testphase ohne Probleme. Allerdings ist eine hohe Rechenleistung notwendig. Auf meinem sehr leistungsstarken Hauptcomputer läuft das System einwandfrei, schwächere Computer bekommen Probleme mit der Echtzeit Erkennung. Um das Videomaterial möglichst hochwertig für die Auswertung durch den Computer bereitzustellen, habe ich mit verschiedenen Bauformen des BienenCounter-Vorbaus sowie unterschiedlichen Kameras (Hersteller) experimentiert.

Fazit und Ausblick

Meine Systeme zeigen erfolgreich, wie technische Hilfsmittel in der Landwirtschaft helfen können neue Aufgaben zu bewältigen. Beim Frühstück zu Hause mal eben nach den Bienen zu schauen, oder von unterwegs, ohne persönlich anwesend sein zu müssen, ist eine deutliche Vereinfachung. Die Kontrolle der Tiere, ohne stören zu müssen, ist für die Tiere viel schonender. Die Ergebnisse aus meinem Projekt können auch in andere Bereiche übertragen werden. Technik ermöglicht nicht nur die Fernüberwachung von Tieren, auch Automatisierungen z.B. beim Fütterungsprozess sind möglich. Die Sensorik kann auch zur Bestimmung von z.B. Bodenfeuchte oder Bearbeitungsfähigkeit von Feldern genutzt werden. Es können Warnmeldungen ausgelöst werden, die räumlich unabhängig empfangen werden können. Der Einsatz von dieser Technik kann z.B. auch in der Pflege oder bei der Überwachung von Säuglingen sinnvoll sein. Generell kann eine Verwendung überall dort denkbar sein, wo Sensorik auf Handlungsbedarf trifft.

Der BienenCounter kann nach entsprechendem Training des Models eine automatisierte Erkennung und Zählung von Tieren, insbesondere Insekten, zu Forschungszwecken durchführen. Die erhobenen Daten könnten dann mittels LoRaWAN in das Labor übermittelt werden.

In der Zukunft möchte ich auf jeden Fall am BienenCounter weiterarbeiten, um die benötigte Rechenleistung zu mindern bzw. die Computer entsprechend ausrüsten. Außerdem könnte der Raspberry Pi AI HAT+ oder ein NVIDIA Jetson (Nano) eine Möglichkeit sein, um effizientere Auswertungen des Videofeeds zu ermöglichen.

Abbildungsverzeichnis

Abbildung 1	Platine BienenHUB V.1.0.1 (Foto: Jan Altenburg).....	4
Abbildung 2	Verschiedene Ansätze für den BienenHUB (erstellt mit draw.io).....	5
Abbildung 3	Solarpanel mit Haltesystem (Foto: Jan Altenburg).....	6
Abbildung 4	Bestückte Platine BienenHUB V3.2 im Gehäuse; Endversion (Foto: Jan Altenburg).....	6
Abbildung 5	Schaltplan der Platine (Screenshot aus KiCad).....	7
Abbildung 6	Deckel und Gehäuse Verbindung (Screenshot aus Fusion 360).....	7
Abbildung 7	Vorbau am Bienenstock (Foto: Jan Altenburg).....	8
Abbildung 8	Konstruktion des Vorbaus (Screenshot aus Fusion 360).....	8
Abbildung 9	BienenCounter Visuelle Ausgabe (Screenshot aus der Visuellen Ausgabe von PyCharm).....	9
Abbildung 10	Ablauf der Datenverarbeitung beim BienenHUB (Erstellt mit draw.io).....	10
Abbildung 11	Ablauf der Datenverarbeitung beim BienenCounter (Erstellt mit draw.io)	11
Abbildung 12	Automatisierung zur Schwarm- und Räubererkennung (Screenshot aus der Home Assistant Web GUI).....	12
Abbildung 13	BienenDashboard1 (Screenshot aus Handy App).....	13
Abbildung 14	BienenDashboard2 (Screenshot aus Handy App).....	13
Abbildung 15	BienenDashboard3 (Screenshot aus Handy App).....	13

Literaturverzeichnis

- [1] „Wie wählt man eine Wägezelle aus? Zu bewertende Faktoren; LAUMAS Elettronica S.r.l.
- [3] [16](https://www.shop.wolf-waagen.de/Bienenstockwaage-ApiGraph-4; 01.01.2025; Bienenstockwaage ApiGraph 4; Wolf Waagen GmbH & Co. KG

</div>
<div data-bbox=)

Unterstützungsleistungen

In Memoriam an Herrn David-Andreas Bergens, CEO bei Pix Software, 2. Projektbetreuer:

Ich bin ohne technisches Vorwissen in dieses Projekt gestartet. Er hat mich immer mit technischem Support unterstützt. Alles Grundwissen, welches ich für den BienenHUB und zur Übertragungstechnik benötigt habe, habe ich von ihm gelernt. Vielen Dank!

Außerdem möchte ich mich bei Herrn Arne Schirmacher, CEO bei Pix Software, für den 30-stündigen KI-Kurs bedanken. Hier hat er unter anderem mir die Basics für die Arbeit mit eigenen KI-Programmen beigebracht. Auf diesem Kurs basiert meine Arbeit zum BienenCounter.

Und zum Schluss möchte ich mich bei meinem Projektbetreuer Herrn Christian Croonenbroek, Lehrer der Gesamtschule Brüggen bedanken. Denn dieser hat immer einen guten Rat parat und liest meine Arbeit Korrektur.