

HoneyPi – Smarte Stockwaage für Imker

Javan Rasokat, Melissa Spinuzza, Kevin Lauff, Maximilian Freisler

Zusammenfassung

Für den Imker ist eine intelligente Stockwaage eine Erleichterung seiner Arbeit, denn ein solches Messsystem liefert vielerlei hilfreiche Ergebnisse. So kann der tägliche Flugbetrieb anhand der Gewichtsänderungen gemessen werden. Ebenfalls wird damit ersichtlich, zu welchen Uhrzeiten die Bienen die meisten Pollen und Honig suchen. Der Honigertrag kann anhand der Gewichtsänderung zum Vortag gemessen werden. Außerdem sagt die Brutraumtemperatur im Winter aus, wann von der Königin noch gebrütet wird. Bei einer plötzlichen Gewichtsabnahme kann ein möglicher Diebstahl des Bienenvolkes erkannt werden. Überdies kann ein Frühwarnsystem Auskunft darüber geben, ob eine große Anzahl an Bienen, z. B. ein ausziehender Bienenschwarm, den Bienenstock verlassen haben.

Das Embedded-Device soll über die angeschlossenen Sensoren verschiedene Messdaten sammeln und über das MQTT-Protokoll an einen zentralen Webserver übertragen. Von dort aus können die Messergebnisse über eine REST-Schnittstelle abgerufen und in einer Smartphone-Applikation grafisch aufbereitet für den Imker dargestellt werden.

Motivation

Eine Marktanalyse ergab, dass Stockwaagen für Imker keine Neuheiten sind und dennoch zwischen 800,00 und 1.000,00 Euro kosten. Zu diesen hohen Anschaffungskosten je Bienenstock, kommen weitere Gebühren für den laufenden Betrieb hinzu, beispielsweise Kosten für die Servermiete, die von den Anbietern veranschlagt werden.

Mit dem HoneyPi-Projekt sollen Imker ihre Stockwaage für unter 150,00 Euro an reinen Materialkosten selbst zusammenbauen können. Die Software und Firmware soll schließlich von www.honey-pi.de, eventuell in einem monatlichen Abo-Modell, erworben werden können.

Architektur

- **Benutzeroberfläche:**

Über eine Android-App sollen die Messdaten für den Imker aufrufbar sein. Darin sollen die Messergebnisse für Temperatur, Luftfeuchtigkeit, Luftdruck und Luftqualität außerhalb des Bienenstocks sowie die Temperatur und das Gewicht des Brutraums je Bienenstock in Diagrammen visualisiert werden. Die Android-App wird in Android-Studio in der Programmiersprache Java entwickelt. Anfänglich wird zur Speicherung der Messdaten die IoT-Plattform ThingSpeak verwendet. Hierüber können auch Benachrichtigungen in Form von E-Mails versendet werden, z. B. bei einer abrupten Gewichtsveränderung. Auch die Versendung von Push-Benachrichtigungen durch die Android-App sind denkbar.

- **Internetschnittstellen:**

Vom Raspberry Pi 3 Model B sollen die Messwerte über das MQTT-Protokoll an ThingSpeak gesendet werden. ThingSpeak bietet von Haus aus bereits geeignete MQTT- und REST-APIs, um Daten hin und zurück zu senden.

- **Zusätzliche Hardware:**

Für die Messungen werden spezielle Sensoren benötigt. Für eine erste Version sind der Temperatursensor DS18S20 für die Messung der Brutraumtemperatur, eine Waage, die am Bienenstock angebracht wird und der BME680-Sensor zur Messung von Temperatur, Luftfeuchtigkeit, Luftdruck und Luftqualität außerhalb der Bienenvölker vorgesehen. In späteren Versionen kann das Projekt auch durch zusätzliche bzw. umfangreicheren Sensoren, beispielsweise die Verwendung des DHT22, der die Temperatur und Luftfeuchtigkeit misst, anstelle des DS18S20, verwendet werden.

Die Übertragung der Messwerte ins Internet sollte über ein Surfstick erfolgen, da nicht alle Imker ein WLAN-Netzwerk an ihren Bienenständen haben. Auch ein leistungsstarker Akkumulator bzw. eine Powerbank, um den Raspberry Pi und die angeschlossenen Sensoren mit genügend Strom zu versorgen, wird benötigt, da der Bienenstock möglicherweise weit von einer Steckdose entfernt steht. Für eine erste Version werden ein oder mehrere leistungsstarke Powerbanks verwendet. Für spätere Versionen des HoneyPi-Projekts könnte für die Stromversorgung auch auf Solarmodule oder eine Kombination aus Powerbanks und Solarmodulen zurückgegriffen werden.

Mit weiteren ESP32-Boards oder Raspberry Pi Zero Ws könnte eine vergleichsweise günstigere Variante entworfen werden, um mehrere Bienenstöcke mit einer Stockwaage

auszustatten. Der Raspberry Pi Model 3 B agiert dabei als Haupteinheit, um die von den Nebeneinheiten gemessenen Daten zu sammeln und an ThingSpeak zu übertragen. Die Kosten für ein ESP32-Board bzw. Raspberry Pi Zero W liegen bei ca. 10,00 Euro, während ein Raspberry Pi Model 3 B für jeden weiteren Bienenstock mit etwa 35,00 Euro zu Buche schlagen würde. Das ESP32-Board bzw. der Raspberry Pi Zero W beinhalten ein WiFi-Modul und GPIO-Pins, an die die Sensoren angeschlossen werden können. Falls für die Nebeneinheiten Raspberry Pi Zero Ws verwendet werden, muss für die Gewichtsmessung zusätzlich ein A/D-Wandler verwendet werden. Der ESP32 hat bereits ein A/D-Wandler integriert.

- **Zu Programmieren**

- **Haupteinheit** (Raspberry Pi Model 3 B):

- An der Haupteinheit müssen der Außensensor BME680, der Brutraumsensor DS18S20 und die Bauteile für die Waage angebracht werden. Die Haupteinheit liest schließlich die eigenen Sensordaten aus, empfängt die Messdaten der optional angeschlossenen Nebeneinheiten und überträgt diese über das ressourcensparende MQTT-Protokoll an ThingSpeak. Deshalb müssen für die Haupteinheit Skripte für das Auslesen und Versenden der verschiedenen Sensordaten und die Verbindung mit den Nebeneinheiten implementiert werden.

- **Nebeneinheit** (ESP32 bzw. Raspberry Pi Zero W):

- An jede optionale Nebeneinheit, die einen weiteren Bienenstock mit dem HoneyPi-Projekt vernetzt, muss lediglich der Innentempersensordaten DS18S20 und die Waage angeschlossen werden. Anschließend liest das Board die Sensordaten aus und überträgt diese über Bluetooth oder WiFi an die Haupteinheit. Demzufolge müssen bei den Nebeneinheiten lediglich Skripte für das Auslesen des DS18S20 und des Wägesensors sowie für das Versenden dieser Daten an die Haupteinheit geschrieben werden.

- **Wartungsmodus:**

- Die Haupteinheit muss in einen Wartungsmodus geschaltet werden können, in dem der Imker über einen Webbrowser eine einfache Konfigurationsseite aufrufen kann. Über diese muss er seine ThingSpeak-Daten zum Senden der Sensordaten und eventuell weitere sensorspezifische Eigenschaften eingeben bzw. auswählen. Auch das Zuordnen von Nebeneinheiten über diese Seite ist sinnvoll. Der Anwender muss bei der Einrichtung in seinem ThingSpeak-Konto beachten, dass er für jeden weiteren

Bienenstock einen weiteren Channel in ThingSpeak erstellt. Außerdem sollte dieser die sogenannten Fields der ThingSpeak-Channel, die jeweils eine Messwertart darstellen, nach einer entsprechenden Namenskonvention benennen, um dies in der Android-Applikation richtig erkennen und darstellen zu können. Dadurch wird gewährleistet, dass die Sensordaten dem richtigen Bienenvolk und der richtigen Messdatenart zugeordnet werden.

Eine Lösung für den Wartungsmodus, um die Konfiguration durchzuführen, könnte eine Art öffentlicher WLAN-Hotspot sein, den der Imker beim Betätigen einer Taste für einen bestimmten Zeitpunkt aktivieren kann. Danach soll schließlich das Öffnen einer Konfigurationsseite im Webbrowser möglich sein, über die die Einstellungen und auch das Zuordnen von Nebeneinheiten möglich sind.

- **Benutzeroberfläche:**

Eine in Java realisierte Android-App, die die Messdaten für den Imker übersichtlich darstellt, dient als Benutzeroberfläche für den Imker. Darin sollen die Messergebnisse für Temperatur, Luftfeuchtigkeit, Luftdruck und Luftqualität außerhalb des Bienenstocks sowie die Temperatur und das Gewicht des Brutraums je Bienenstock in Diagrammen visualisiert werden. Anfänglich wird zur Speicherung der Messdaten die IoT-Plattform ThingSpeak verwendet. Aus diesem Grund muss der Benutzer auch beim erstmaligen Start der Applikation seinen User-API-Key von ThingSpeak eingeben.

Hardware-Skizze

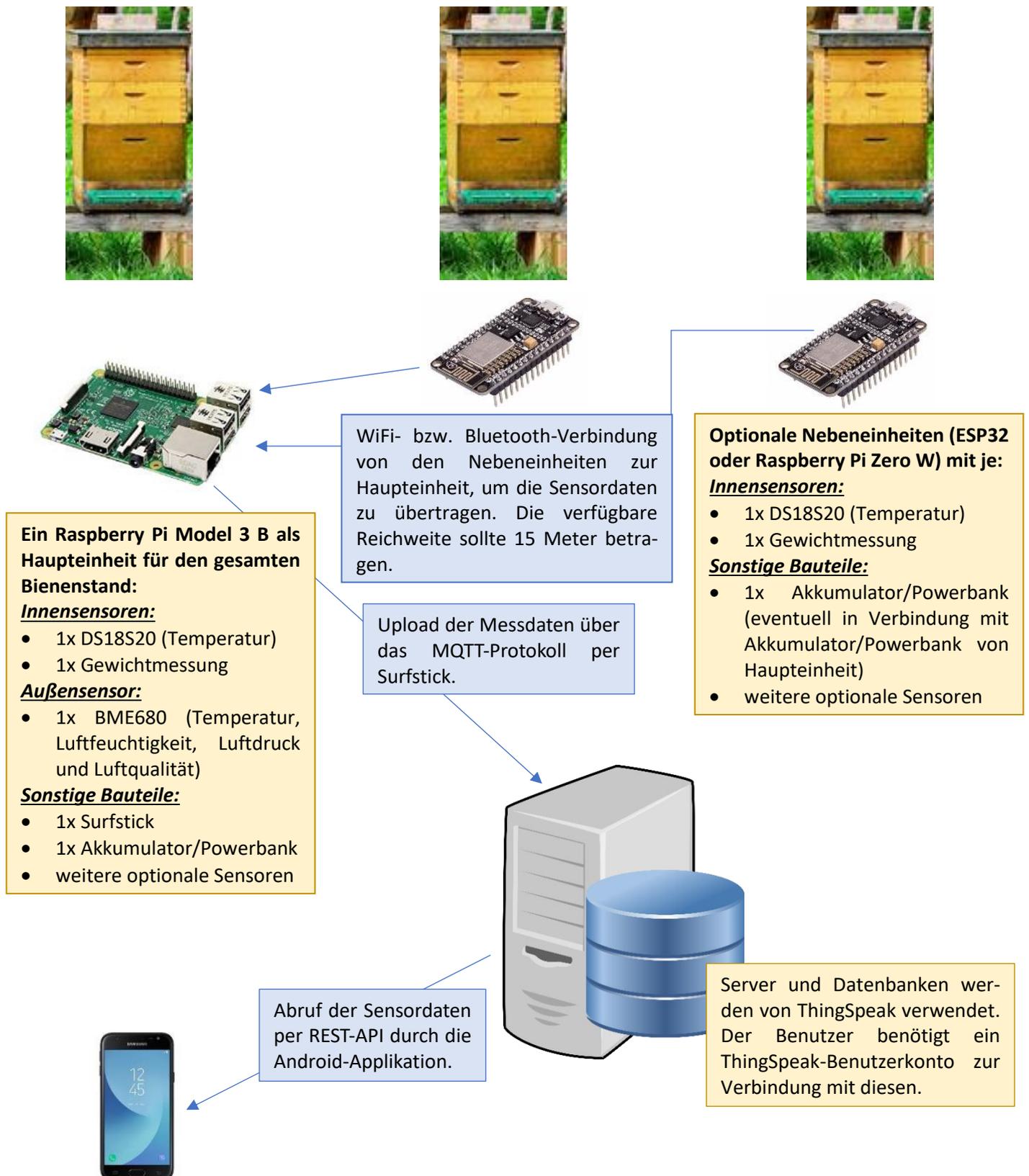


Abbildung 1: HoneyPi-Architektur

Stückliste

Die Stückliste wird für drei Bienenstöcke veranschlagt. Als Board für die Nebeneinheiten wird hier von einem ESP32 ausgegangen. Dadurch entfallen die A/D-Wandler für die Gewichtssensoren (Wägezellen). Die Versandkosten der einzelnen Positionen wurden nicht berücksichtigt.

Tabelle 1: Stückliste für das Projekt

Nr.	Rubrik	Beschreibung	Lieferant	Artikelnummer	Anz.	Preis/St.
1	Systemboard (Haupteinheit)	Raspberry Pi 3 Model B	Rasppishop	eb6598	1	34,51 €
2	Internet	Huawei E3531 SurfStick	notebooksbilliger	A 617934	1	24,95 €
3	Speicher	Sandisk microSD 4GB Class4 für die Haupteinheit	Rasppishop	eb5785	1	3,99 €
4	Systemboard (Nebeneinheiten)	Espressif ESP32	eBay	253059783728	2	9,45 €
5	Sensor	DS18S20 1-Wire- Temperatursensor	Pollin Electronics	180014	3	1,95 €
6	Widerstand	4,7 kΩ Widerstand für den DS18S20	eBay	191872425388	3	0,023 €
7	Sensor	Wägezellen je 50 kg Maximallast (4er-Pack)	Amazon	B00R1J7VA0	3	10,59 €
8	A/D-Wandler	HX711 A/D-Wandler für die Waage	eBay	152972823886	1	2,49 €
9	Sensor	BME680-Sensor	PIMORONI	PIM323	1	21,00 €
10	Jumper-Kabel	Female-Female Jumper Kabel 20 cm	Amazon	B078L9QZZJ	17	ca. 0,012 €
11	Powerbank	30.000 mAh Powerbank	Amazon	B06WP7C8ZK	3	45,99 €

Mensch-Maschine-Schnittstelle

Das HoneyPi-Projekt bietet zwei Mensch-Maschine-Schnittstellen: eine Android-App, welche Messdaten visualisiert, und eine Konfigurationsseite, über welche Einstellungen vorgenommen werden können.

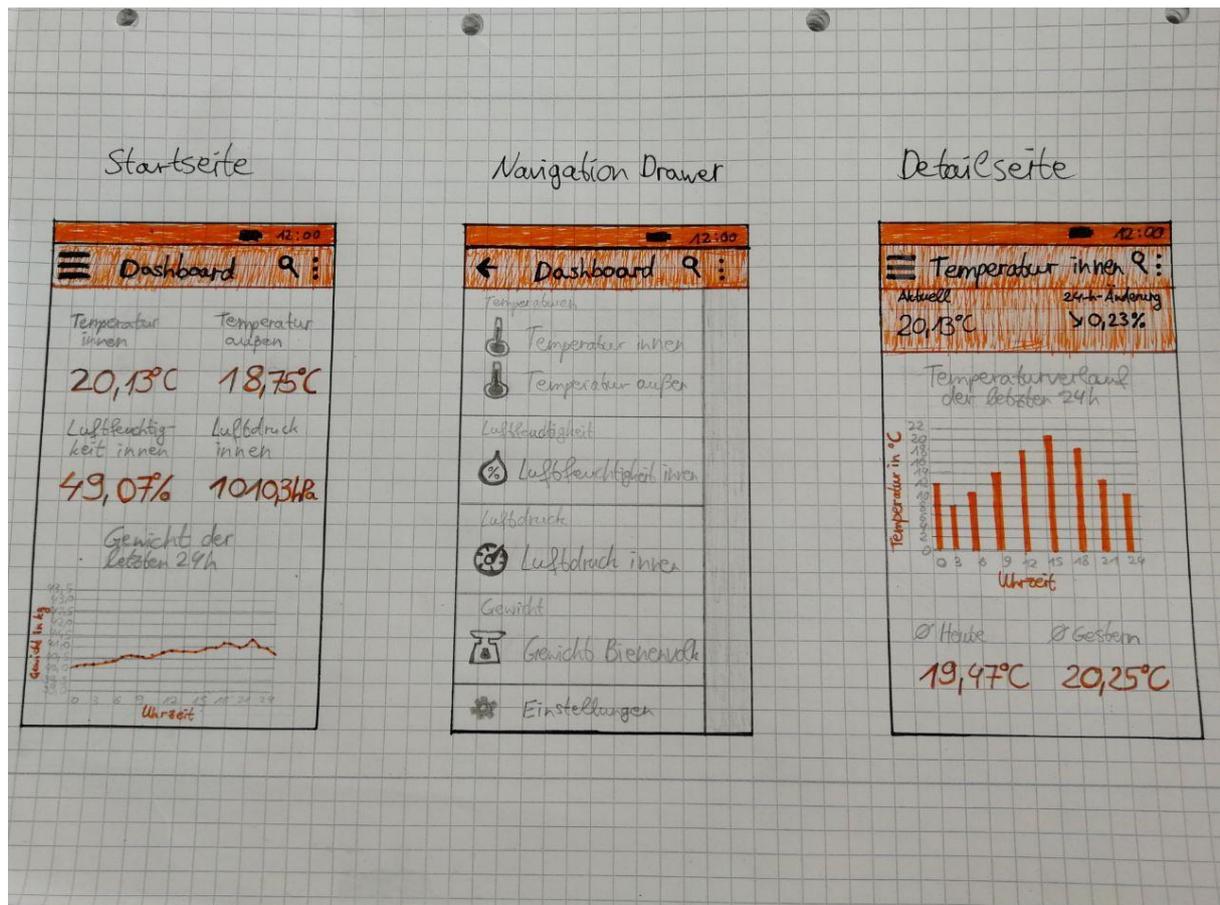


Abbildung 2: Erste Mockups der Android-Applikation

Die Android-Applikation:

Die Android-App bietet auf ihrer Startseite ein Dashboard, auf dem eine Zusammenfassung über sämtliche Werte dargestellt werden. Über ein Navigation-Drawer-Menü auf der linken Seite kann ein spezifischer Wert zur näheren Betrachtung ausgewählt werden. Auf einer solchen Detailseite werden dann neben dem aktuellen Messwert auch Durchschnittswerte, Verlaufsdigramme und die Messhistorie angezeigt. Die Messdaten werden über eine REST-API vom ThingSpeak-Konto des Benutzers abgerufen. Deshalb ist es auch notwendig, dass dieser beim erstmaligen Start der Applikation seinen User-API-Key von ThingSpeak eingibt.

Die Konfigurationsseite:

Die Haupteinheit kann in einen Wartungsmodus geschaltet werden, in dem der Benutzer eine Konfigurationsseite über einen Webbrowser aufrufen kann. Über diese Seite ist es dem Anwender möglich, seine ThingSpeak-Daten einzugeben, an die die Messwerte der Sensoren gesendet werden sollen. Außerdem soll es ihm dort möglich sein, weitere sensorspezifische Einstellung vorzunehmen sowie Nebeneinheiten dem gesamten System zuzuordnen.

Bestandteile der Firmware

Tabelle 2: Bestandteile der Firmware des Projekts

Package	Zweck
Linux Kernel	Grundgerüst der Firmware
BusyBox	Shell, Init-System und einfacher Webserver
hostapd	Wifi Access Point Deamon
dhcpcd, dhclient	DHCP-Server und Client
python3	Interpreter für Python-Programme
php	Zur Verwendung von PHP-Programmen
openssh	Zum Aufbauen von SSH-Verbindungen
sudo	Befehle mit Super-User-Rechten ausführen
git	Quellcode von GitHub herunterladen
wchar	Um nano verwenden zu können
nano	Editor zum Bearbeiten von Dateien
wiringpi	GPIO-Bibliothek, die insbesondere für die I2C-Pins benötigt wird

Für die Firmware wird als Grundgerüst der Linux-Kernel verwendet. Dieser wird mit dem Werkzeugkasten BusyBox erweitert. BusyBox bietet unter anderem eine Shell, das Init-System und auch einen einfachen Webserver, über den die Konfigurationsseite gestartet werden soll. Dadurch entfällt eine zusätzliche Installation eines Webserver-Packages.

Um die Konfigurationsseite in einem Webbrowser zu erreichen, soll sich der Anwender mit einem beliebigen Endgerät über einen WLAN-Access-Point mit dem Raspberry Pi verbinden können. Hierfür werden die Packages hostapd, dhcpcd und dhclient verwendet. Da die Konfigurationsseite in PHP geschrieben werden soll, wird das gleichnamige Package benötigt.

Weitere Skripte, insbesondere zum Auslesen der Sensoren, werden in Python programmiert. Deshalb ist auch der Python-Interpreter für die Firmware notwendig.

Um sich mit einem entfernten Computer über SSH mit dem HoneyPi-System verbinden zu können, wird das Package openssh verwendet. Um Befehle mit Super-User-Rechten auszuführen, wird sudo verwendet.

Da es bereits Bibliotheken, die in unserem Projekt verwendet werden können, vorhanden sind, wird git installiert, um diese leicht von GitHub herunterladen zu können.

Zum einfachen Editieren von Dateien, wird der Editor nano und seine benötigte Toolchain wchar integriert.

Wiringpi ist eine Bibliothek, um die GPIO-Pins des Raspberry Pi einfacher zu erreichen. Dies ist insbesondere für die zwei I2C-Pins wichtig, die für den BME680-Sensor benötigt werden.