

HTW Berlin

Angewandte Informatik (B.Sc)

Dozent

Mobile Anwendungen, Technik Mobiler Systeme

Prof. Dr. Huhn, Alexander

Wintersemester 2018/19

Projektdokumentation

Implementierung eines Multi-Hop Netzwerkes

unter Verwendung von Long Range Modulen

Autor

Michael Schwabe (558507)

Inhaltsverzeichnis

1.	Einleitung und Motivation.....	3
2.	Grundlagen.....	3
2.1	Long Range Modul (LoRa Modul)	3
2.2	UART (Universal Asynchronous Receiver Transmitter)	3
2.3	Raspberry Pi.....	4
3.	Konzepte und Aufbau	4
3.1	Konzept Protokoll	4
3.2	Verkabelung	7
4.	Umsetzung und Implementierung	7
5.	Ausblick und Fazit	10
6.	Quellen / Literaturverzeichnis	11
7.	Anlagen.....	11
7.1	Repository	11
7.2	Abbildungen	12

1. Einleitung und Motivation

Dieses Projekt soll veranschaulichen, wie ein Multi-Hop Netzwerk grundsätzlich funktioniert und das dieses auch mit den verwendeten Komponenten betrieben werden kann. Hierbei können die Vorteile der LoRa Module ausgeschöpft werden. In dem Kapitel Grundlagen wird weiter auf die Module eingegangen. Des Weiteren wird in diesem Projekt auch die Kommunikation mit den Modulen behandelt und die Implementierung des Protokolls. Implementierung soll innerhalb des ISM (Industrial, Scientific and Medical) Bandes stattfinden und Frequenzen 433,05 MHz bis 434,79 MHz nutzen.¹

2. Grundlagen

2.1 Long Range Modul (LoRa Modul)

Das verwendete LoRa Modul HIMO-01M (das M steht für Modul) sitzt auf einer I/O Adapterplatine HIMO-01P (das P steht für Platine). Das drahtlose Übertragungsmodul, HIMO-01M basiert auf dem drahtlosen Transceiver SX1278 von SEMTECH. Die Herstellerangaben besagen, dass mit fortschrittlicher LoRa-Spread-Spectrum-Technologie (Frequenzspreizung) Kommunikationsentfernungen von 10.000 Metern erreicht werden. Diese Größen, wie auch die anderen Spezifikationen und Leistungsangaben, können in diesem Projekt nicht oder nur bedingt geprüft werden.²

Dank des Mitgelieferten integrierten Modul MCU STM8L051F3, kann das Modul direkt mit AT Befehlen (seriell) gesteuert werden.³

Aufgrund, der durch SEMTECH patentierten Modulationstechnik verfügt das Modul über eine erhöhte Sensitivität von -148 dBm bei einer Ausgangsleistung von +20 dBm (100mW), einer großen Übertragungsdistanz und einer hohen Zuverlässigkeit.⁴ Diese Werte sind Spezifikationen vom Hersteller und sollten bei dem Einsatz in kritischen Infrastrukturen genauer untersucht bzw. belegt werden.

2.2 UART (Universal Asynchronous Receiver Transmitter)

„Universal Asynchronous Receiver Transmitter“ dient zur Realisierung digitaler serieller Schnittstellen. In dem Projekt wird dieses Bauelement in Verbindung mit einem USB –

¹ <https://de.wikipedia.org/wiki/ISM-Band> , aufgerufen am 12.01.2019

² <https://eckstein-shop.de/HIMALAYA-I-O-Adapter-Board-mit-LoRa-Modul-MCU-SMA-Buchse-433MHz> , aufgerufen am 12.01.2019

³ <https://eckstein-shop.de/HIMALAYA-I-O-Adapter-Board-mit-LoRa-Modul-MCU-SMA-Buchse-433MHz> , aufgerufen am 12.01.2019

⁴ <https://eckstein-shop.de/HIMALAYA-I-O-Adapter-Board-mit-LoRa-Modul-MCU-SMA-Buchse-433MHz> , aufgerufen am 12.01.2019

Adapter eingesetzt. Eine bildliche Darstellung befindet sich in Abschnitt Anlagen 7.2 Abbildungen, Abbildung 1. Für die korrekte Verwendung sollten RX und TX gekreuzt werden. Im Kapitel Konzepte und Aufbau wird dies dargestellt.

2.3 RaspberryPi

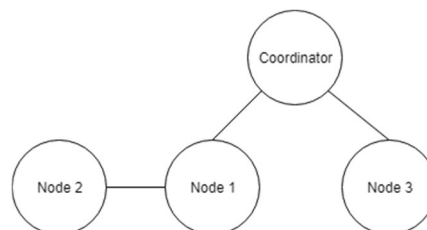
Der RaspberryPi (3B+) dient in diesem Projekt als Umgebung zum Entwickeln und Testen. An diesem werden auch die Module bzw. später nur ein Modul via UART – Schnittstelle (USB-Schnittstelle seriell) angesprochen. Zur Entwicklung dient die Thonny IDE mit Python 3 und als Versionsverwaltung Git.

3. Konzepte und Aufbau

3.1 Konzept Protokoll

Das Protokoll soll einen einheitlichen Netzwerkaufbau gewährleisten. Des Weiteren sollen Knoten sich in dem Netzwerk einheitlich kommunizieren. Dazu muss ein Koordinator angesprochen werden, um den Knoten feste Adresse zu vergeben.

Abb. 3 Beispiel von Assoziationsbeziehungen in dem Netzwerk



Anhand der Abbildung 3 kann die Assoziationsbeziehung zwischen den Knoten in diesem Netzwerk dargestellt werden. Alle Knoten, auch der Koordinator, sollen sogenannte „Full-Function-Devices“ sein. Das Attribut besagt, dass alle Geräte alle Funktionen übernehmen können.

So kann Node 2 Koordinator sein wie Node 3. In dem Konzept wird ein Koordinator (nur einer) vorgesehen, der wie bereits angedeutet Knoten registriert bzw. feste Adressen aus einem speziellen Adressenpool zuweist (Tabelle 1).

Tabelle 1: Adressräume und reservierte Adressen

Adresse bzw. Adressraum	Anwendung
0000	Koordinator
0001 – 0010	Für zukünftige Zwecke reserviert
0011 – 0FFF	Adressen für neue Knoten (temporär)
1000 – FFFE	Feste Adressen (ca. 61400) durch Koordinator zugewiesen
FFFF	Broadcast

Die Assoziationsbeziehungen in Abbildung 3 besagen auch, dass Node 2 mit Node 3 kommunizieren kann, obwohl diese sich nicht „sehen“ bzw. physikalisch drahtlos in Verbindung stehen.

Dies soll durch die anderen Teilnehmer bewerkstelligt werden, welche die Nachrichten weiterleiten. Somit entsteht ein Multi-Hop Netzwerk. Aus zeitlichen Gründen wurde ein Routingverfahren im klassischen Sinne nicht vorgesehen. Es handelt sich um ein reines Weiterleiten der Nachrichten.

CODE	MESSAGE ID	TTL	HOPS	SRC ADDR	DST ADDR
PAYLOAD					

Abb. 4 Headerkonzept⁵

Diese Nachrichten werden so wie in Abbildung 4 aufgebaut. Alle Nachrichten werden entsprechend einen Message-Code (CODE) besitzen. Anhand dieses Codes kann die Nachricht einem speziellen Event zugeordnet werden und von jedem Knoten rasch, präzise und zweckgebunden verarbeitet werden. In Tabelle 2 sind die implementierten Message-Codes übersichtlich dargestellt.

Zusätzlich wird die Nachricht mit einer individuellen Identifikationsnummer ausgewiesen. Diese dient dazu, Nachrichten nicht unendlich weiterzuleiten. Wenn man ein Echo einer bereits weitergeleiteten Nachricht erhält, wird diese Nachricht nicht weitergeleitet, da sie bereits weitergeleitet wurde. Somit werden Echos und Schleifen vermieden. Ergänzend dazu ist noch der Time-To-Live (TTL) Counter eingebunden. Bei 0 wird das Paket dann verworfen und nicht weitergeleitet. HOPS kennzeichnet die aktuellen Sprünge bzw. Weiterleitungen. Quell und Zieladresse identifizieren die Kommunikationspartner.

⁵ <https://github.com/HTW-LowRateNet/Protocol/blob/master/Protokoll%20f%C3%BCr%20Multihop-Netzwerk%20V4.pptx>, aufgerufen am 12.01.2019

Tabelle 2: Nachrichten- / Message-Codes

CODE	PAYLOAD	DST Adresse	Bedeutung
CDIS	0 Bytes	0000	Coordinator Discovery (neuer Knoten versucht Koordinator zu finden)
ADDR	4 Bytes	Temporäre Adresse	Ist die Antwort auf eine Adressanfrage (vom Koordinator)
ADDR	0 Bytes	0000	Ist die Adressanfrage an einen Koordinator (vom neuen Knoten)
AACK	0 Bytes		Quittung der Adressanfrage für den Koordinator
MSSG	Beliebig	beliebig	Nachricht an Knoten im Netzwerk (registrierte Knoten)
ALIV	0 Bytes	FFFF	Kontinuierlicher Heartbeat (jede Minute) des Koordinator und die Antwort auf eine CDIS Nachricht.
NRST	0 Bytes	FFFF	Network-Reset, dies geschieht nur wenn der Koordinator ein ALIV bekommt

Wie in Tabelle 2 zu entnehmen ist, sind dort bereits schemenhaft Muster zu erkennen, wer mit wem in welchen Zustand kommuniziert. So ist der Message-Code CDIS dafür vorgesehen, Koordinatoren zu finden bzw. aufzufordern, einen ALIV zu senden. Dies hat die Bewandnis, dass die Koordinatoren nur periodisch auf sich aufmerksam machen. Erstens, um das Netzwerk nicht mit Nachrichten zu fluten und zweitens, weil es nicht effektiv und effizient ist, die Intervalle kürzer 60 Sekunden anzusetzen (60 Sekunden sind bereits zu kurz, Empfehlung wäre Delta größer 300 Sekunden).

Damit wird den neuen Knoten eine Möglichkeit gegeben, sich zu registrieren. Denn erst nach dem die Knoten eine feste Adresse vom Koordinator erhalten haben, dürfen diese in dem Netzwerk wirklich aktiv werden. Vorher ist es den temporären Adressinhabern untersagt Nachrichten weiterzuleiten. Wenn ein neuer Knoten auf einen CDIS ein ALIV erhält, sendet er sofort mit ADDR eine Anfrage nach einer festen Adresse.

Wenn der Koordinator die Anfrage erhält, sendet er wiederum auch eine ADDR-Nachricht, mit einer freien Adresse aus dem Adresspool (des Koordinators). Der anfragende Knoten entnimmt dem PAYLOAD der Nachricht die Adresse und richtet diese als seine aktuelle ein. Mit der neuen Adresse sendet der Knoten ein AACK (Acknowledge) dem Koordinator. Der Koordinator weiß nach dem Empfang der AACK-Nachricht, dass der Knoten die Adresse bekommen hat und dass der Knoten diese auch eingerichtet hat. Grund ist, dass die Nachricht als Quelladresse die vergebene Adresse hat.

Wenn auf CDIS Nachrichten einer bestimmten Anzahl keine ALIV-Nachrichten folgen, macht sich der Knoten selber zum Koordinator. Dieser wird jetzt wie bereits beschrieben auf die Message-Codes reagieren. Zusätzlich reagiert der Koordinator selber auf ein fremdes ALIV mit einem NRST (Network-Reset). Dieser Message-Code dient der

Reorganisation des gesamten Netzwerkes. Jeder, der diesen Message-Code erhält, setzt sich selber zurück und beginnt wieder mit einem CDIS. Damit wird vermieden, dass es mehr als einen Koordinator gibt. Spezifische Implementierungen folgenden in dem Kapitel Umsetzung und Implementierung sowie in dem Git-Repository unter Anlagen zu finden.

3.2 Verkabelung

Folgende Verkabelung muss hergestellt werden.

- X1 zu Y1 (3.3V zu VIN)
- X2 zu Y2 (GND zu GND)
- X3 zu Y3 (TX zu RX)
- X4 zu Y4 (RX zu TX)
- X5 dient am HIMO-01P dem „Chip-Reset“

In den Anlagen unter 7.2 Abbildungen befindet sich die Abbildung 1 und Abbildung 2.

Diese bilden, zu der beschriebenen Verkabelung, eine Optische Ergänzung. Ein falsches verkabeln kann zu erheblichen irreparablen Schäden an den Geräten führen.

4. Umsetzung und Implementierung

Das Protokoll wurde in diesem Projekt mit der Programmiersprache Python umgesetzt.

Implementiert wurden zwei Klassen und ein Skript, welches die Klassen initialisiert und entsprechend auf den Input reagiert. Abbildung 4 stellt das Klassendiagramm dar.

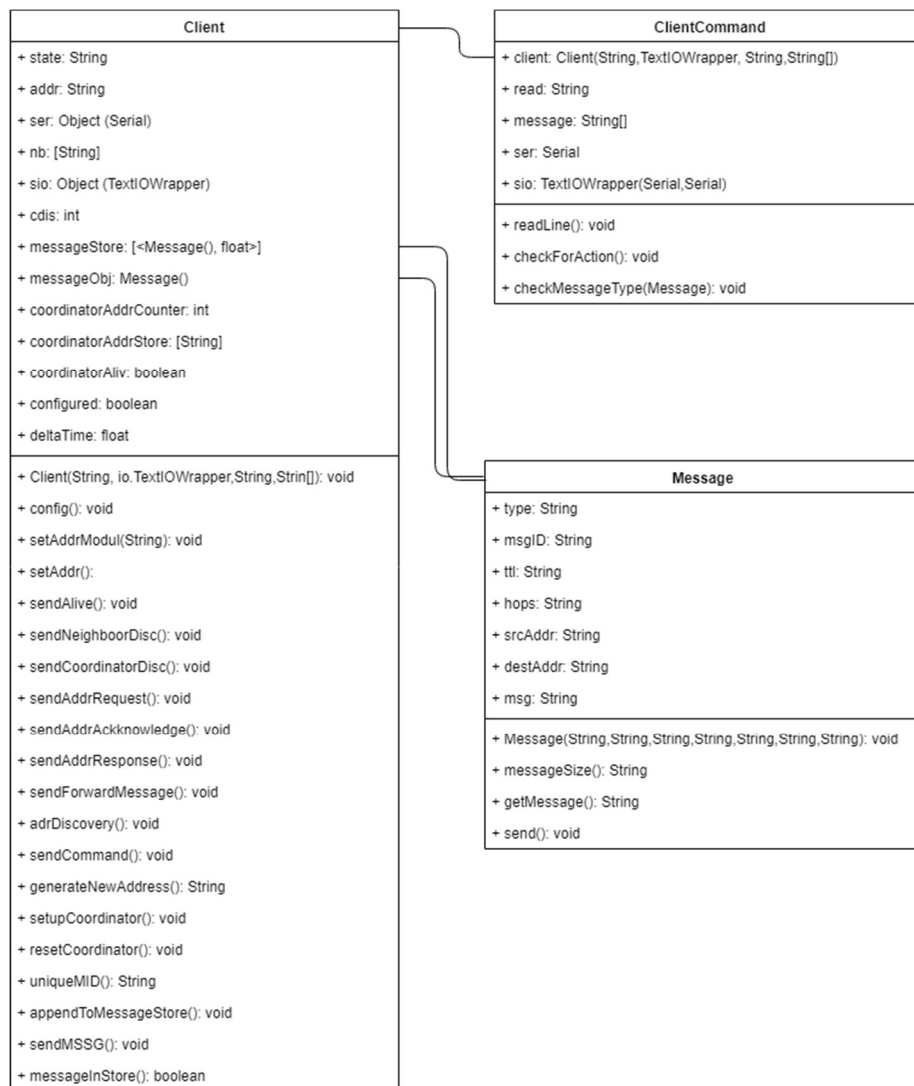
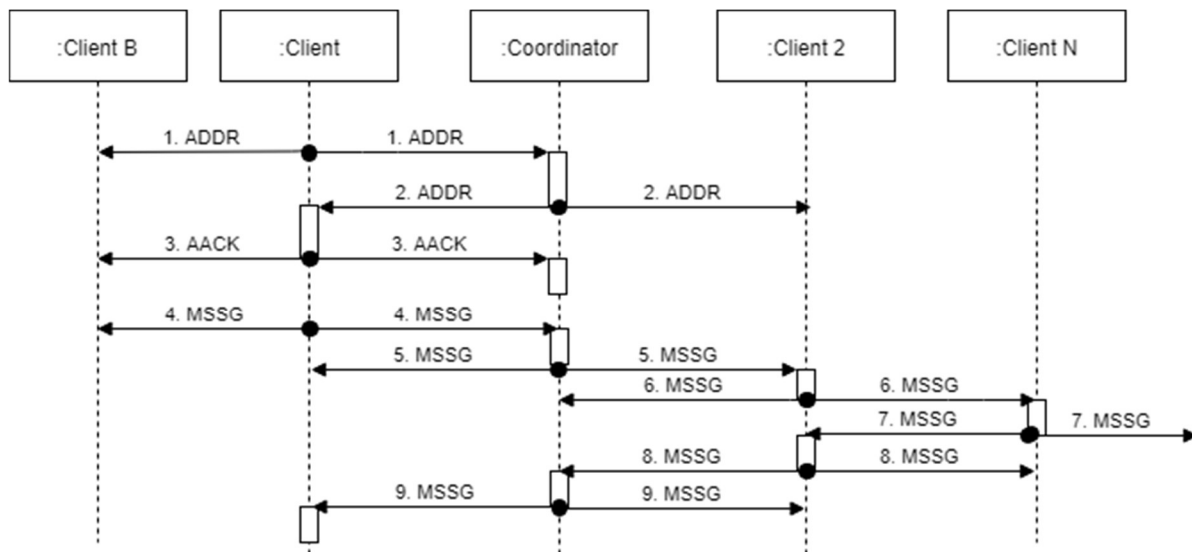


Abb. 4 Klassendiagramm des Projektes

Hier können entsprechende Assoziationen abgeleitet werden. Eine klassische Vererbung findet nicht statt. Grund dafür ist der Einsatz einer schlanken Semaphore. Die Semaphore „state“ regelt entsprechend die Aufgaben bzw. weist ein Client Objekt als CL, COOR oder NEW aus. Die drei Stadien des Betriebes entscheiden maßgeblich über das Handling von Nachrichten.



Beispiel Inhalt der einzelnen Nachrichten während der Adressvergabe und eventuellen Nachrichtenaustausch zwischen Knoten mit entsprechendem Forwarding/Weiterleitung.

1. ADDR, 000100, 100, 0, SRC ADDR (Client), DST ADDR (0000), PAYLOAD=0
2. ADDR, 000101, 100, 0, SRC ADDR (0000), DST ADDR (Client), PAYLOAD=NEW ADDRESS
3. AACK, 000102, 100, 0, NEW ADDRESS (Client), DST ADDR (0000), PAYLOAD=0
4. MSSG, 000110, 100, 0, SRC ADDR (Client), DST ADDR (Client N), MESSEGETEXT
5. MSSG, 000110, 100, 1, SRC ADDR (Client), DST ADDR (Client N), MESSEGETEXT
6. MSSG, 000110, 100, 2, SRC ADDR (Client), DST ADDR (Client N), MESSEGETEXT
7. MSSG, 000150, 100, 0, SRC ADDR (Client N), DST ADDR (Client), MESSEGETEXT
8. MSSG, 000150, 100, 1, SRC ADDR (Client N), DST ADDR (Client), MESSEGETEXT
9. MSSG, 000150, 100, 2, SRC ADDR (Client N), DST ADDR (Client), MESSEGETEXT

Abb. 5 Sequenzdiagramm Adressvergabe, Nachrichtenaustausch und Weiterleitung

Das spezifische Handling kann in Abbildung 5 nachvollzogen werden. Wenn ein Client, wie in Abbildung 5, dem Coordinator ein ADDR (1.ADDR aus Abbildung 5) Paket sendet, wird dies auf dem Broadcast getan und jeder Teilnehmer in Reichweite erhält dieses Paket. Es dürfen nur und ausschließlich Nachrichten von Clients weitergeleitet werden die den Status CL oder COOR besitzen. Somit sind Clients, mit dem Status NEW oder einem anderen Status, keine vollwertigen Teilnehmer des Netzwerkes. Diese Trennung hat zur Laufzeit große Vorteile und dient einem geregelten Netzaufbau.

Besonders zu erwähnen ist, die MessageQueue bzw. der messageStore[<Message,float>] aus der Abbildung 4. Dieser Speichert bereits versendete und weitergeleitete Nachrichten. Das ist essenziell. Echos oder Weiterleitungen eigener ALIV Nachrichten würden für einen Network-Reset sorgen. Des Weiteren wird ein PingPong-Effekt vermieden und die Parameter HOPS und TTL könnten Sinnvolle Rückschlüsse zulassen.

5. Ausblick und Fazit

Zusammenfassend ist zu sagen, dass die Zeit ausreichte, um grundlegende Funktionalitäten eines sich selbstorganisierenden Netzwerkes zu implementieren. Jedoch fehlen höherwertige Funktionalitäten, um das Netzwerk und die Komponenten effektiv und effizient kommunizieren zu lassen. Auch der Einsatz eines einzigen Koordinators sollte überdacht werden. Auch die theoretische Teilnehmerzahl von ca. 60.000 Knoten würde das Netzwerk mit dem Protokoll nicht verkraften. Die Erweiterung mit einem Routingverfahren (Distanzvektor verfahren - AODV) sollte angestrebt werden.

Schemenhaft ist in diesem Projekt ein enormes Potenzial zu sehen, welches weiter ausgebaut werden kann und sollte. Szenarien für den urbanen Einsatz würden sich aufgrund des verwendeten Frequenzbereichs (ISM Band von 433,05 MHz bis 434,79 MHz) sehr gut eignen. Szenarien wie die aktuelle Position der öffentlichen Verkehrsmittel die in einem Multi-Hop Netzwerk ihre aktuelle Position (Fahrgäste oder Infotafel an den Haltestellen) sowie deren Zustand für die Verkehrsbetriebe übermitteln, um die Wartung zu optimieren. Diese Übermittlung sollte aber nicht auf die illusorischen 10.000 Meter angedacht werden. Die Entfernung könnte auf 500 Meter bis hin zu 3.000 Meter angedacht werden, wo sich die nächste Haltestelle befindet.

Ein Einsatzszenario im heterogenen Umfeld wäre die Auswertung bzw. Übertragung entlang von Wasserstraßen wie die Spree, Elbe oder Main, um den Wasserstand aktuell zu prüfen. So könnten, bei Schmelz oder gefährlichen Hochwasser, frühzeitig, automatisch und im optimalen Verhältnis Schleusen geöffnet oder andere Vorsichtsmaßnahmen des Zivil- und Katastrophenschutzes ergriffen werden. Dies kann zur Vermeidung oder Prävention von hohen Sach- und Personenschäden führen. In diesem Bereich könnte ein Einsatz sogar notwendig sein, da die Technologie bzw. die Geräte mit ihrer Beschaffenheit ideal wären. Aufgrund der Kosten und der Verfügbarkeit sowie der Robustheit ist ein redundanter Aufbau und Einsatz sinnvoll und auch erschwinglich für Länder und Kommunen.

Die Szenarien können von der Waldbrand Registrierung bis hin zu Wettervorhersagen fortgeführt werden. Was mit den Szenarien aufgezeigt werden sollte, ist ein universeller Einsatz von Multi-Hop Netzwerken. Gleichzeitig wurde aufgezeigt das dies mit den, in diesem Projekt verwendeten Geräten auch sinnvoll und umsetzbar ist. An der Effizienz und Effektivität des Protokolls kann und muss noch weitergearbeitet werden.

Zu bedenken ist, dass der Einsatz in kritischen Infrastrukturen gesondert geprüft werden sollte, da die Kommunikation entsprechend abgesichert werden muss. Hier kann eine Verschlüsselung des Inhaltes angedacht werden. Aber auch eine Transportverschlüsselung wäre denkbar was jedoch erhöhten Aufwand und eventuell neue Komponenten benötigen würde. Der Schlüsselaustausch würde nicht über das Netzwerk, sondern an den Komponenten direkt geschehen, um eine gewisse Kapselung für die kritische Infrastruktur zu schaffen. Die Verschlüsselung des Inhaltes vor der Übertragung würden dann wiederum andere Komponenten (ComputeModul RaspberryPi, Industrial SoC von Intel-Apollo, Allwinner etc.) übernehmen.

6. Quellen / Literaturverzeichnis

7. Anlagen

7.1 Repository

Der Code ist unter dem folgenden privaten Repository:

<https://github.com/HTW-LowRateNet/s0558507>

Entwürfe, Konzeptionelle Ideen (Routing) und Skizzen befinden sich ebenfalls in dem Repository.

Ausarbeitungen der gesamten Projektgruppe (Abbildung 4,5 und 6) befinden sich in folgendem Repository:

<https://github.com/HTW-LowRateNet/Protocol>

7.2 Abbildungen

Abbildung 1 UART USB Adapter



Abb. 1 UART USB Adapter

Abbildung 2 HIMO-01P Platine

HIMO-01P

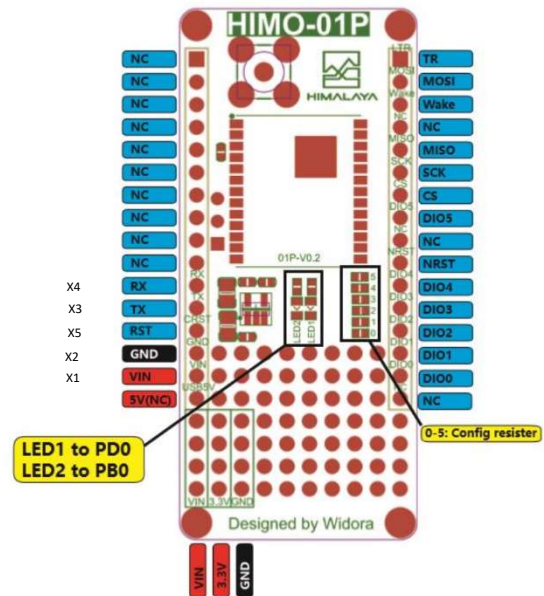


Abb. 2 HIMO-01P Platine⁵

⁶ <https://eckstein-shop.de/HIMALAYA-I-O-Adapter-Board-mit-LoRa-Modul-MCU-SMA-Buchse-433MHz> , aufgerufen am 12.01.2019