

Velo-Trackingsystem Logbuch

Projektunterricht 2021 G3b

Aaron Kuhn, Francois Kieninger, Piotr Dzianach, Tom Ritter

Datum	Was und Wer
22.08	<p>Recherche welche Teilchen für das Trackingsystem gebraucht werden. Zur Übertragung der Daten scheint das Mobilfunknetz die beste/einfachste Lösung zu sein.</p> <ul style="list-style-type: none"> • Als Microcontroller ein esp32, da er billig und mächtig ist • Als GPS den Neo-6m, da ich schon zwei zu Hause habe • Sim800I für Mobilfunk, wird oft in Tutorials verwendet und einfach handzuhaben, leider 2G (habe aber keine alternativen gefunden) <p>Die Anzahl an Teilchen müsste noch bestimmt werden und danach müsste man diese kaufen. Der Kauf sollte demnächst gemacht werden, da die Lieferzeiten recht lang sind.</p>
25.08	<p>Wir haben besprochen, dass wir Teilchen für 4 Fahrräder kaufen. Falls wir Fahrräder bei Bahnhöfen aufstellen, sollte dies gut genug sein. Mehr Fahrräder würden wahrscheinlich unser Budget sprengen und es wäre viel mehr Arbeit die Fahrräder zu finden/reparieren, ausstatten und zu managen. Diese Zahl sollte aber nicht unbedingt ein Ziel sein, sondern mehr eine Abschätzung um die Teilchen Kaufen.</p>
27.08	<p>Heute haben wir die Teilchen gekauft, welche wir am 22.08 erwähnt haben. Wir haben auf die Teilchen auf Aliexpress gekauft, nachdem wir verschiedene Anbieter verglichen habe.</p> <ul style="list-style-type: none"> • Esp32 • neo-6m • Sim800I <p>Die Teilchen sollten irgendwann in den Herbstferien ankommen.</p>
28.08	<p>Wir haben versucht das GPS-Modul mit einem esp32 devkit zum Laufen zu Bringen. Um die nach dem NMEA Standard kodierte Nachrichten vom GPS-Modul zu dekodieren, benutzen wir die Bibliothek TinyGPS++. Leider haben wir es nicht geschafft unsere Position zu ermitteln. Nach recht langen Fehlersuchen ist uns aufgefallen, dass die übergebenen Daten des GPS-Moduls leer waren. Nach einer Überprüfung ist uns aufgefallen, dass der Sensor mit keinem Satelliten verbunden war. Auch nachdem wir ihn 1 Stunde lang draussen rumliegen hatten, konnte er sich nicht verbinden.</p>
29.09	<p>Das GPS-Modul hat es geschafft sich mit Satelliten zu verbinden, nachdem wir es weitere 2 Stunden draussen liegen hatten. Jetzt braucht es nur noch ein paar Minuten, um sich mit den Satelliten zu verbinden.</p>
12.09	<p>Wir haben zu Hause zwei mehr oder weniger vollständige Fahrräder am Stehen, wo wir für das Projekt verwenden könnten. Wir haben uns die beiden Fahrräder angeschaut. Wir hatten leider keine Zeit mehr sie aufzupumpen, um mit ihnen zu fahren und so die Gangschaltung und die Schläuche zu testen. Unsere Erkenntnisse habe wir in einem Dokument eingetragen.</p>
17.09	<p>Heute sind zwei Pakete aus China eingetroffen. Eines beinhaltet die Neo-6m GPS Module. Das andere Paket enthält die Sim800I GSM Module. Leider ist uns da aufgefallen, da die Bestellung für die Esp32 vom Verkäufer abgebrochen wurde und somit diese nie ankommen werden. Dies stellt für uns ein Problem dar, da die Lieferzeit über 1 Monat ist und wir je nachdem dann nicht mehr genug Zeit haben die Elektronik rechtzeitig fertig zu haben. Wir müssen uns jetzt noch überlegen, wo wir die Esp32 kaufen.</p>
22.09	<p>Wir haben unser Vertrag ausgebessert, er sollte jetzt fertig sein. Wir müssen ihn nur noch ausdrucken und unterschreiben. Wir haben auch die Rollen aufgeteilt:</p> <ul style="list-style-type: none"> • Recherche: Aaron • Velos: Piotr

	<ul style="list-style-type: none"> Elektronik und Server: Tom & François
20 – 25.09	<p>Wir haben uns die Datenübertragung angeschaut. Dabei kommen für uns 2 Protokolle in Frage: http oder mqtt. Wir haben die Vor- und Nachteile auf einem Blatt festgehalten. habe zu beiden Protokollen Testprogramme geschrieben, in denen ich Daten von meinem Laptop zu einem Lokalen Raspberry Pi geschickt habe.</p> <p>Wir haben aber noch ein grosses Problem was jegliche Kommunikation betrifft. Wir brauchen eine IP-Adresse, auf welche wir für unseren Server benutzen können. Sonst ist die Datenübertragung unmöglich. Bei mir zu Hause weiss ich, dass es nicht möglich ist eine statische IP-Adresse zu bekommen. Wir müssen eventuell bei den anderen schauen.</p>
25.09	<p>Francois: Heute bin ich bei der Abfall-Sammelstelle beim Technischem Zentrum von Obersiggenthal vorbeigegangen, um zu schauen, ob es im Altmetall eventuell ein altes Fahrrad hat. Ich habe dort 2 fast vollständige Fahrräder gefunden. Mit diesen Velos und denen von mir zuhause haben wir 4 Velos, was unser Ziel ist. Wir müssen aber noch alle reparieren.</p>
26.09	<p>Francois: Ich habe mir die Stromversorgung vom System angeschaut. Die Schwierigkeit hier ist, dass der Esp32 3 - 3.6 Volt braucht und das GSM-Modul 3.4 - 4.4 Volt braucht. Im Internet wird aber empfohlen den Sim800l mit mindestens 3.8 Volt zu versorgen, da er sonst nicht zuverlässig funktioniert. Bei Transmission Bursts kann der Sim800l bis zu 2 A ziehen. In solchen Situationen geht die Spannung herunter. Deshalb empfiehlt der Hersteller auch den Sim800l mit 4V zu versorgen. Die Idee war 3 Alkali-Mangan-Zellen (voll haben diese 1.5V) Seriell zusammenzuschliessen damit wir eine Spannung von 4.5 Volt haben. Damit der Esp32 aber nicht verbraten wird kommt zwischen ihm und den Batterien ein Spannungswandler, der die Spannung auf 3.3V reduziert. Die 4.5 sollten mit dem Sim800l schon gehen, da die Batterien über die Zeit Spannung verlieren. Leider ist mir auf der Suche nach einem Spannungswandler aufgefallen, dass diese eine sogenannte Drop-Out Voltage haben. Das heisst sie brauchen einen bestimmten Unterschied zwischen der Input Voltage und der Output Voltage. Die tiefste Spannung für 3.3V wäre 4.5V gewesen. Da die Batterien aber Spannung verlieren funktioniert es nicht.</p> <p>Der Lösungsansatz zu diesem Problem ist 4 Batterien Seriell zusammenzuschliessen und somit 6V zu bekommen. Dies genügt für den Spannungswandler. Für den Sim800l zapfen wir bei der 3. Batterie den Strom ab und haben somit maximal 4.5 Volt, was hoffentlich nicht zu viel ist.</p> <p>Dies ist nur eine theoretische Überlegung. Es wäre gut, wenn man dies testen könnte. Es wäre auch interessant sich diese Peaks in der Stromstärke an einem Oszilloskop anschauen könnte.</p>
29.09	<p>Francois: Ich habe mir angeschaut, wie wir am besten den Zeitpunkt in der Datenbank speichern. Meine Lösung sieht folgendermassen aus: Wenn die Informationen in die Datenbank eingespielen werden, speise ich auch den derzeitigen Unix timestamp ein. Dieser wird dann in der Datenbank umgewandelt und als den Typ datetime gespeichert.</p>
30.09	<p>Francois: Heute habe ich mir angeschaut, wie man eventuell die Applikation machen kann. Ich habe zwei Arten gefunden, wie man die App verwirklichen kann. Man kann eine schon existierende Open-source Karten App nehmen und diese für unseren Zweck Modifizieren. Hierbei ist das Problem, dass man sich recht stark mit dieser App</p>

	<p>auseinandersetzen muss um sie, zu verstehen damit man sie dann modifizieren kann. Hier ist mir ein interessanter Link aufgefallen: https://www.geopaparazzi.org/#/ . Die Zweite Art und Weise wäre mit dem Own map feature von Google Maps zu arbeiten. Hier könnte man diese Karte mit der Google API verändern. Mit dieser Methode bekommt man eine Webapplikation, welche aber auch als App verpackt werden kann. Leider habe ich noch nicht ganz verstanden, wie es mit der Google API funktioniert (wie viele Zugriffe man hat etc.). Wie man das machen wird, können wir entscheiden, wenn wir die App machen müssen.</p>
02.10	<p>Francois: Ich habe mir nochmal angeschaut, wie man eventuell ein App machen könnte. Dieses Mal habe ich nach Bibliotheken gesucht, um eine Karte in eine App zu integrieren. Ich habe verschiedene open Source Bibliotheken entdeckt:</p> <ul style="list-style-type: none"> • mapsforge • Tangram <p>Es gibt weitere closed source Bibliotheken, die ich nicht unbedingt benutzen würde, da das freie Abo nur wenig kann:</p> <ul style="list-style-type: none"> • here maps • Google Maps • mapbox <p>Bisher denke ich, dass mapsforge am besten ist, da es auch offline funktioniert und frei ist.</p>
03.10	<p>Francois: Ich habe mir den GPS-Sensor nochmals angeschaut. Er hat wieder einige Stunden gebraucht, bis er mit genügen Satelliten verbunden war. Ich habe auf dem Internet nach Leuten gesucht, die das gleiche Problem haben, habe aber nichts gefunden. Meine Vermutung ist, dass das GPS-Modul sich die Position der Satelliten merkt, damit es schneller geht, um sich wieder zu verbinden. Nun ist aber glaube ich die kleine Batterie am Modul leer/kaputt und er merkt sich die Sachen nicht mehr. Man muss also noch machen:</p> <ul style="list-style-type: none"> • GPS-Sensor draussen testen, vielleicht hat er einfach Mühe, weil er in einem Gebäude ist. • Ich habe bisher immer das gleiche Modul genommen. Ich sollte mal ein anderes nehmen. Dieses Modul haben wir zu Hause schon seit einiger Zeit. Vielleicht funktioniert eines der neu angekommenen Modelle besser.
04.10	<p>Francois: Ich habe mit kicad ein Schaltbild von der Elektronik gemacht. Ich habe ein anderes GPS-Modul genommen und versucht, ob es ein Signal bekommt. Ich habe es ungefähr eine Stunde draussen liegen lassen und es hatte immer noch kein Signal. Ich lass es über die Nacht draussen liegen und schau dann, ob es sich verbunden hat.</p>
05.10	<p>Francois: Das neue GPS-Modul, welches ich gestern Abend nach draussen gestellt hat, konnte keine Verbindung zu Satelliten machen.</p> <p>Nach einiger Recherche ist mir aufgefallen, dass ich nicht der einzige bin, bei dem die Module nicht funktionieren haben. Ich bin in den Rezensionen auf Aliexpress auf zwei Personen gestossen, bei welchen auch die GPS-Module nicht funktionierenden. Sie weisen darauf hin, dass bei ihnen der Antennenanschluss Falsch herum angelötet war. Bei mir ist dies bei 3 von den 4 neuen GPS-Modulen der Fall. Um diese zu reparieren, muss man den Anschluss entlöten, drehen und dann wieder hin löten. Dies wird aber nicht allzu simpel sein, da der Anschluss recht klein ist.</p>

	<p>Das vierte, mit korrektem Antennenanschluss, GPS-Modul hat sich aber trotzdem nicht mit Verbunden. Ich habe versucht herauszufinden, ob es vielleicht in einem Power Safe Modus ist. Leider habe ich nicht geschafft dies zu ermitteln.</p> <p>Ich habe auch angefangen mich mit dem UBX-Protokoll auseinanderzusetzen. Mit diesem Protokoll kann ich mit dem GPS-Modul kommunizieren, um es in Sleep Modus zu versetzen. Wahrscheinlich werde ich dazu eine eigene Bibliothek schreiben müssen. Zum Glück hat mein Bruder für seine Matura eine solche Bibliothek für den Raspberry Pi geschrieben. Ich kann mich somit Teils auf diese Bibliothek stützen.</p> <p>Ich habe mir auch angeschaut, wie es sich mit dem Speicher verhält. Bisher verwendet ein Programm ca. 20% des Speicherplatzes. In solch einem Programm sind aber auch ein Haufen Standard Bibliotheken enthalten. Die GSM-Bibliothek braucht nach dem Entwickler ca. 50% des Speicherplatzes. Somit wären noch 30% frei, was glaube ich noch reichen wird.</p> <p>Um zusammenzufassen was man von heute noch fertigmachen muss:</p> <ul style="list-style-type: none"> • UBX, eine leichte Bibliothek finden oder selber eine schreiben • GPS-Module, Antennen umdrehen • Herausfinden warum die Antennen nicht funktionieren
06.10	<p>Francois: Heute habe ich mich weiter mit dem UBX-Protokoll auseinandergesetzt. Ich werde wahrscheinlich den Treiber von meinem Bruder auf den Arduino portieren und die Bibliothek mit Power-Management Commandos ergänzen. Ich habe angefangen die Struktur von der Bibliothek zu verstehen. Den Teil, den ich neu schreiben muss, ist der uart (Universal asynchronous receiver/transmitter) Teil. Dies muss ich komplett neu mit Funktionen für den esp32 neu schreiben. Ich habe mich auch mit dem Protokoll an sich selbst auseinandergesetzt und habe geschafft ein Programm zu schreiben welches eine Vorgefertigte Nachricht and das GPS-Modul schickt. Ich muss aber noch etwas schreiben zum Sehen, ob wirklich etwas zurückkommt.</p> <p>Ich habe auch versucht das GSM-Modul mit dem esp32 zu verbinden. Leider ist der esp32 immer abgestürzt und hat eine "Guru Meditation Error" von sich gegeben. Zum Glück konnte ich dieses Problem lösen. Das Programm hat Funktionen vom Atmega328P benutzt. Ich habe das Programm so geändert, dass es Funktionen von der EspSoftwareSerial Bibliothek benutzt und somit kompatibel mit dem esp32 ist. Leider habe ich nicht geschafft, dass das GSM-Modul mit dem esp32 kommuniziert. Das GSM-Modul benutzt eine andere Spannungsversorgung als der esp32, da es ca. 4 Volt braucht und dies zu viel für den esp32 ist.</p>
07.10	<p>Francois: Ich habe geschafft den esp32 mit dem GSM-Modul kommunizieren zu lassen. Das Problem lag bei den Spannungsversorgungen. Da die beiden keinen gemeinsamen Ground hatten konnten diese nicht über Pins kommunizieren. Das Problem war recht schnell gelöst, indem ich die beiden Grounds verbunden habe.</p> <p>Ich habe weiter daran gearbeitet mit dem GPS-Modul per UBX zu kommunizieren. Leider konnte ich keine wirklichen Erfolge machen. Ich konnte bisher keine UBX-Nachricht empfangen.</p> <p>Ich habe versucht bei einem GPS-Modul den Antennenanschluss umzudrehen. Da dieser sehr klein ist, ist es recht schwierig. Ich habe es noch nicht geschafft.</p>
08.10	<p>Francois: Ich habe weiter versucht den Antennenanschluss umzudrehen. Es ist mir leider noch nicht ganz gelungen ich konnte aber einen Pin fast ganz von Lötzinn befreien. Ich muss mir mal anschauen, wie man korrekt so kleine Bauteile entlötet.</p>

	<p>Mir ist auch eingefallen, wie wir entdecken können, ob sich jemand auf dem Fahrrad befindet. Der esp32 hat sogenannte touch-Pins. Diese bemerken es, wenn sich etwas mit einer elektrischen Ladung sich ihnen nähert. Der Mensch ist leicht elektrisch geladen und somit bemerkt dieser sensor es. Wahrscheinlich kann man ein Kabel in den Sattel legen und dies mit einem solchen touch-Pin verbinden. Wenn sich nun jemand auf den Sattel setzt, kann man dies erkennen.</p>												
12.10	<p>Francois: Heute habe ich angefangen mir die Struktur des definitiven Programmes zu überlegen. Ich habe auch angefangen den Teil mit dem GPS zu schreiben. Die Struktur des GPS sieht folgendermassen aus:</p> <ol style="list-style-type: none"> 1. Esp32 wake-up 2. Neo-6m wake-up 3. Neo-6m data collection 4. Neo-6m sleep 5. Sim800l wake-up 6. Sim800l send data 7. Sim800l sleep 8. Esp32 sleep <p>Wobei ich hier noch nicht weiss, ob ich das GPS-Modul neo-6m zum Schlafen bringe, da ich dazu glaube ich das UBX-Protokoll verwenden muss und ich noch nicht weiss, ob ich es schaffen werde es zu benutzen.</p>												
14.10	<p>Francois: Ich habe das richtig gelötete neue GPS-Modul zum Laufen gebracht. Dazu habe ich es einfach mit der Antenne vom alten GPS-Modul verbunden. Es hat sich in unter einer halben Stunde mit genügend Satelliten verbunden.</p> <p>Ich habe mir auch angeschaut was für einen Spannungsregler ich nehmen soll. Die meisten Spannungsregler, die ich gesehen habe, hatten eine Dropout Voltage von ca. 1.2 V. Das heisst die Batterien müssen zusammen mindesten 4.5 V liefern damit der Spannungswandler 3.3 V liefern kann. Alkaline Batterien verlieren aber Spannung, wenn sie sich entladen. Ich habe mit 4 Batterien gerechnet:</p> $4 * 1.12 \text{ V} = 4.48 \text{ V} \cong 4.5 \text{ V}$ <p>Das heisst bei einer Dropout-Voltage von 1.2 V braucht jede Batterie ca. 1.12 V. Nach dieser Tabelle von Wikipedia, können die Batterien nur um so ca. 50% entladen werden. Darum ist meine Wahl auf den LM2937 gefallen. Dieser hat eine tiefere Dropout-Voltage. So um die 0.7 V. Dies heisst die Spannung der Batterien darf auf so ca. 4.0 V runtergehen.</p> $4 * 1.0 \text{ V} = 4 \text{ V}$ <p>Dies heisst die Batterien können wahrscheinlich bis auf 10% bis 20% entladen werden was um einiges mehr ist. Ich habe das mit kicad erstellte Schaltbild aktualisiert, indem ich den LM2937 eingesetzt habe.</p> <p>Weiter habe ich mich mit den Antennen für das GPS-Modul auseinandergesetzt.</p> <table border="1"> <thead> <tr> <th>Patch-Antenne</th><th>Resultate</th></tr> </thead> <tbody> <tr> <td>Alte, 0</td><td>Funktioniert</td></tr> <tr> <td>1</td><td>Nicht geschafft</td></tr> <tr> <td>2</td><td>Nicht geschafft</td></tr> <tr> <td>3</td><td>Konnte verbinden, Resultate fraglich: 4.669518,37.255809 (Äthiopien), sonst nicht mehr geschafft</td></tr> <tr> <td>4</td><td>Funktioniert</td></tr> </tbody> </table> <p>Ich konnte noch nicht alle austesten, dies muss ich noch morgen fertig machen.</p>	Patch-Antenne	Resultate	Alte, 0	Funktioniert	1	Nicht geschafft	2	Nicht geschafft	3	Konnte verbinden, Resultate fraglich: 4.669518,37.255809 (Äthiopien), sonst nicht mehr geschafft	4	Funktioniert
Patch-Antenne	Resultate												
Alte, 0	Funktioniert												
1	Nicht geschafft												
2	Nicht geschafft												
3	Konnte verbinden, Resultate fraglich: 4.669518,37.255809 (Äthiopien), sonst nicht mehr geschafft												
4	Funktioniert												

15.10	<p>Francois: Heute habe ich weiter die GPS-Antennen getestet und die Resultate in die Tabelle eingetragen. Was vielleicht ein Störfaktor war, ich habe die GPS-Module mit der Antenne auf unsere Loggia gestellt. Es kann sein, dass das Haus das Signal gestört hat. Man müsste die Antennen aufs Dach oder auf eine freie Fläche stellen, damit man optimale Resultate bekommt. Ich glaube nicht, dass es so viel ausmacht und die Antennen sollen ja auch in nicht optimalen Verhältnissen funktionieren. Meine Schlussfolgerung ist einfach, dass diese Antennen wahrscheinlich defekt sind. Ich habe auch ein Abonnement bei Digitec für eine Sim-Karte gelöst. Es ist ein Abo mit unlimitierten Daten und einer maximalen Geschwindigkeit von 0.4 Mbit/s.</p>
16.10	<p>Francois: Ich habe versucht das GSM-Modul in Betrieb zu nehmen. Ich habe wie letztes Mal geschafft in Betrieb zu nehmen und mit dem Modul zu kommunizieren. Dabei habe ich mit den AT Befehlen gearbeitet. Ich habe eine Anleitung befolgt, um es mit dem GPRS zu verbinden. Die APN für diese Sim-Karte von Digitec ist "internet" der Befehl sieht also folgendermassen aus: "AT+CSTT='internet','','"'. Leider habe ich nicht geschafft mich zu verbinden. Die Einstellung haben aber funktioniert. Ich bin mir nicht sicher, warum es nicht funktioniert, ich glaube aber, dass es an mangelndem Signal lag. Ausserdem kenn ich mich mit den AT Befehlen noch nicht so ganz aus und muss mich noch mit der Referenz auseinandersetzen.</p>
17.10	<p>Francois: Heute habe ich die fehlenden Teile bei Reichelt bestellt. Wir haben uns für die folgenden Teile entschieden:</p> <ul style="list-style-type: none"> • Spannungsregler LM2937 • Kondensatoren welche wir für den Spannungsregler brauchen à 0.15 µF und 10 µF • ESP32-WROOM-32E • Steckbrett • Pin-Header
31.10	<p>Francois: Ich habe versucht das GSM-Modul mit dem 2G Netz zu verbinden dieses Mal habe ich es manuell mit den AT Befehlen versucht und mit der Bibliothek TinyGSM (hintenrum benutzt sie auch bestimmte AT befehle). Leider ist es mir nicht gelungen. TinyGSM hat mir die Meldung gegeben das vielleicht etwas mit der CREG Option nicht stimmt. Ich hoffe das dies die richtige Spur ist.</p>
01.11	<p>Piotr & Francois: Wir haben heute angefangen die Fahrräder zu reparieren. Leider ging es nicht so schnell wie wir gehofft hatten. Wir haben geschafft ein Fahrrad komplett zu reparieren. Bei einem muss man nur noch die Bremse reparieren. Den dritten haben wir uns angeschaut und aufgepumpt. Das vierte Velo haben wir uns nicht angeschaut.</p>
06.11	<p>Francois: Ich habe mich nochmal mit dem GSM-Modul befasst. Ich habe versucht herauszufinden, ob das Modul überhaupt ein Netzwerk sieht. Dazu habe ich mit meinem Handy geschaut, ob es dort, wo ich bin, 2G hat. Als dies funktioniert hat, habe ich mich informiert, wie ich die vom GSM-Modul sichtbaren Netzwerke sehen kann. Mit dem Befehl AT+COPS=? Gibt es mir die sichtbaren Netzwerke aus. Man muss aufpassen AT+COPS? Ist nicht das gleiche. AT+COPS=? Gab mir an ein Sunrise Netzwerk zu sehen. Dies heisst die Antenne des GSM-Moduls sollte funktionieren. Ich vermute, dass das Problem an der APN liegt. Wenn ich die APN setze, gibt es mir ein OK zurück. Wenn ich aber die APN Abfrage, bekomme ich die Standard APN ("CMNET","","") oder es kommt gar nichts zurück. In der Dokumentation ist mir auch aufgefallen, dass es verschiedene Arten und weissen gibt die APN zu setzten. Ich muss mich genauer informieren wie die APN funktioniert. Die Bisherigen möglichen APN sind:</p>

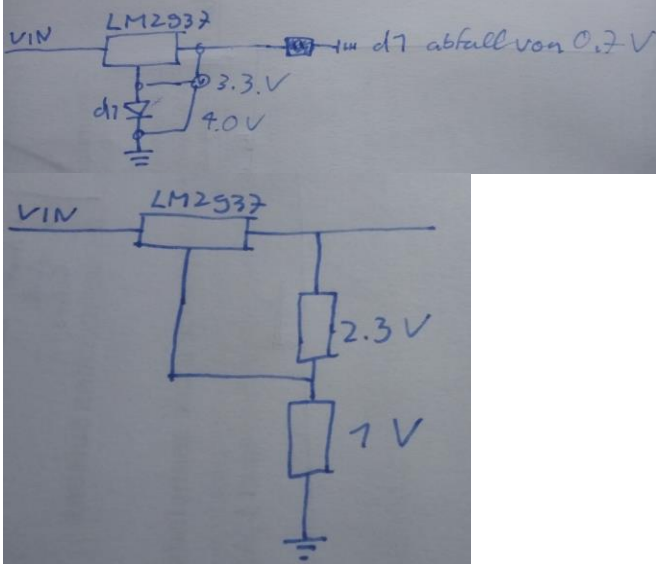
	<ul style="list-style-type: none"> • "internet", "", "" • "internet", "internet", "internet" • "dr.m2mr.ch", "", "" <p>Bisher vermute ich, dass es die zweite APN ist.</p>
10.11	<p>Alle: Wir haben zusammen angeschaut wie weit wir sind und was wir machen müssen. Es wird vielleicht eine bis 2 Wochen Verzögerung bei der Elektronik geben. Sonst läuft noch alles gut.</p>
14.11	<p>Francois: Heute habe ich wieder am GSM-Modul gearbeitet. Ich habe versucht es mit dem 2G Netz zu verbinden. Ich habe es auf manuelle Art und Weise versucht. Dabei bin ich wieder am Punkt mit der APN stecken geblieben. Ich habe verschiedene Befehle versucht und es immer noch nicht geschafft die APN zu ändern. Ich habe aber nicht alle Befehle ganz verstanden, die ich ausprobiert habe. Ich muss mich hier noch mehr informieren.</p> <p>Ich habe es auch mit der TinyGSM Bibliothek ausprobiert. Es hat auch nicht funktioniert. Es gibt mir die Fehlermeldung: "Unhandled: AT+CREG?". Der CREG Befehl gibt den derzeitigen Status des Moduls an (z.B. Verbunden, auf Netzwerksuche etc.). Wenn ich diesen Befehl manuell teste, bekomme ich immer etwas zurück. Ich weiss nicht an was der Fehler liegt.</p>
15.11	<p>Francois: Ich habe mir den AT+CREG? Befehl angeschaut. Dieser gibt an, ob das GSM-Modul mit dem Mobilfunknetz verbunden ist. Normalerweise sollte dies automatisch geschehen, soweit ich verstanden habe. Der Befehl gibt mir aus, dass das Modul nicht angemeldet ist und auf der Suche nach einem Operator ist. Ich sehe bisher zwei Optionen warum das Modul sich nicht verbindet. 1.</p> <ol style="list-style-type: none"> 1. Es fehlen irgendwelche Informationen damit es sich verbinden kann. Ich habe aber online nichts gefunden was ich falsch gemacht hätte. 2. Es liegt an fehlendem Signal. Wir wissen vom AT+COPS=? Befehl, dass das Modul das Netzwerk sehen kann. Es braucht aber immer ein paar Sekunden, bis es etwas antwortet was vielleicht ein Hinweis ist, dass die Verbindung instabil ist. <p>Folgende Punkte wären also gut auszuprobieren, um zu testen, ob es am Signal liegt:</p> <ul style="list-style-type: none"> • Andere GSM-Module benutzen. • Andere Antennen und Antennenarten benutzen. • An einem Ort mit besserer Signalstärke. • Über längere Zeiträume. Bisher habe ich das GSM-Modul nur nebenherlaufen lassen. Dabei lief es maximal 2 Stunden lang ohne, dass es neu gestartet wurde.
17.11	<p>Francois: Ich habe ein drittes GSM-Modul parat gemacht und getestet. Ich hatte die gleichen Resultate wie bei den anderen. Ich habe auch alle Antennen von einem höheren Punkt aus probiert, da man dann meiner Erfahrung nach eine bessere Signalqualität hat. Es hat aber nicht geholfen.</p> <p>Die GSM-Module geben mir nach jedem Befehl die Meldungen "SMS Ready" und "Call Ready" aus. Dies sind sogenannte "unsolicited result codes", kurz URC. Diese beiden werden, nachdem das Modul hochgefahren ist und die demensprechende Funktion initialisiert, wurde ausgegeben. Das heisst sie sollten nicht nach jedem Befehl ausgegeben werden. Daraus kann man schliessen, dass sich das Modul wahrscheinlich immer wieder neustartet. Die Neustarts können meinem Verständnis nach aus folgenden Gründen stattfinden.</p> <ul style="list-style-type: none"> • Eine falsche Einstellung oder Befehl

	<ul style="list-style-type: none"> • Eine nicht genügende Stromversorgung • Das GSM-Modul ist defekt <p>Ich habe mir einmal die Stromversorgung angeschaut. Zur Versorgung des GSM-Moduls verwende ich ein Netzteil, welches bis zu 1 A liefern kann. Das GSM-Modul kann aber bis zu 2 A ziehen. Ich habe aber nochmals den Stromverbrauch gemessen. Das GSM-Modul hat nie mehr als 0.2 A gebraucht. Ich habe es mit einem Messgerät und der Anzeige vom Netzteil gemessen. Vielleicht hat es sehr kurze Peaks, welche zu kurz sind, damit man sie auf den Anzeigen sieht. Ich halte dies aber eher für unwahrscheinlich. Es könnte auch sein, dass die Kabel zwischen dem Netzteil und dem GSM-Modul zu schlecht sind.</p> <p>Darum muss man folgendes noch ausprobieren. Bessere Kabel verwenden und ein Kondensator beim GSM-Modul hinzulöten, um möglich Peaks zu überbrücken.</p>
18.11	<p>Francois: Ich habe mich wieder mit dem GSM-Modul beschäftigt. Ich habe and das GSM-Modul zwei Kabel und einen Kondensator mit 10 μF hingelötet. Die Idee dahinter war, dass die Kabel besser Leiten als die, die ich bisher verwendet habe. Der Kondensator soll helfen, falls es eine Spitze im Verbrauch hat. Die Ergebnisse haben sich aber nicht geändert.</p> <p>Darum habe ich später einen Grösseren Kondensator mit 2200 μF genommen. Auch habe ich nach einem Ratschlag von meinem Vater den Kondensator und die Stromversorgung möglichst nah am Modul zu löten. Nach dieser Modifikation hat das GSM-Modul keine URC mehr ausgegeben. Auch war das Modul viel responsiver. Ich habe geschafft, dass sich das Modul mit dem Netzwerk verbindet und, dass es sich in das GPRS (Mobile Daten) einloggt. Ich konnte meine IP-Adresse ausgeben. Ich habe aber nicht getestet, ob es wirklich eine Internetverbindung hat.</p> <p>Das Problem lag also doch in der Stromversorgung. Wahrscheinlich ist der Stromverbrauch während dem Verbinden mit dem Netzwerk kurz hoch. Dieser Peak ist so kurz, dass man ihn auf dem Messgerät nicht sieht. Dieser Peak ist mehr, als das was das Netzteil liefern kann und deshalb startet sich das GSM-Modul neu. Der Kondensator in diesem Fall entlädt sich während dem Peak und bewahrt das Modul vor dem Neustart.</p> <p>Man muss nun testen ob man es schafft eine Nachricht per MQTT zu senden.</p>
19.11	<p>Francois: Heute habe ich getestet, ob ich es schaffe eine Nachricht per MQTT zu versenden. Dazu habe ich die TinyGSM und die PubSubClient Bibliotheken verwendet. Die PubSubClient Bibliothek implementiert das MQTT Protokoll. Die TinyGSM Bibliothek kommuniziert mit dem GSM-Modul und übernimmt alle Befehle, die ich vorher manuell eingegeben habe. Dazu dient es noch als Modem für die PubSubClient Bibliothek und ermöglicht ihr so den Zugang zum Internet. In einem ersten Schritt habe ich es geschafft anhand eines Beispielsprogramms der TinyGSM Bibliothek mich mit einem Broker zu verbinden und eine Nachricht zu schicken und zu empfangen.</p> <p>In einem zweiten Schritt habe ich am finalen Programm weitergearbeitet und die Schritte integriert, damit ich etwas per MQTT über das GSM-Modul verschicken kann. Es hat alles funktioniert. Als Broker, habe ich einen öffentlichen Broker von HiveMQ benutzt.</p> <p>Die folgenden Schritte sehen Folgendermassen aus:</p> <ul style="list-style-type: none"> • Im Hauptprogramm den Teil mit dem GPS fertig schreiben damit man diese Daten verschicken kann.

	<ul style="list-style-type: none"> • Die ganze Elektronik zusammenstecken und schauen, ob man es schafft seine Koordinaten per MQTT zu versenden • Einen eigenen Broker aufsetzen, da man die öffentlichen Broker nur zu Testzwecken gedacht sind und somit nicht sicher für echte Benutzung sind. Wir haben einen Raspberry Pi auf welchem man das machen könnte
20.11	<p>Francois: Ich habe heute den Teil mit dem GPS fertiggeschrieben. Dazu habe ich die TinyGpsPlus Bibliothek verwendet. Schwierigkeit hierbei war herauszufinden, ob man brauchbare Koordinaten herausgelesen hat. Auch musste man in einem späteren Schritt die Koordinaten vom float Format in ein Array aus char umwandeln, damit man es per MQTT versenden kann. Ich habe einen Umweg über das String Format genommen, da ich es einfacher finde mit dem zu Arbeiten. Eventuell muss man eine andere Lösung finden, da diese nicht sehr sauber ist.</p> <p>Die Daten werden im JSON Format verschickt, da man sie so einfacher verarbeiten kann. Eine Nachricht sieht also folgendermassen aus: {"lat":47.49, "lng":8.29}.</p> <p>Ich habe es Erfolgreich geschafft meine Position per MQTT über GSM zu versenden. Leider hatte ich zwei Mal das Problem, dass der Esp32 und das GSM-Modul nicht geschafft haben miteinander zu kommunizieren. Es ist selten, aber es kann später eventuell zu Problemen führen.</p>
21.11	<p>Francois: Ich habe mich mit dem Sleep-Modus für das GSM-Modul beschäftigt. Meine allererste Idee war das GSM-Modul einfach auszuschalten und wieder anzuschalten, wenn ich es wieder gebraucht habe. Dazu muss man aber Zugang zum sogenannten PWRKEY Pin haben. Bei meiner Version vom Sim800l ist dieser nicht herausgeführt und somit nicht benutzbar. Deshalb habe ich mir den Sleep-Modus angeschaut. Hierzu muss man beim Sim800l den DTR Pin auf Hoch setzten. Um das GSM-Modul wieder aus dem Sleep aufzuwecken, muss man den DTR Pin wieder auf Tief setzten. Das heisst die ganze Zeit während den Sim800l im Sleep ist muss der DTR Pin Hoch sein. Das Problem hier ist, dass ich den Esp32 und den Sim800l gleichzeitig im Sleep haben will. Der Esp32 kann aber im Sleep nicht mit den Pins arbeiten und so den DTR Pin auf Hoch halten. Ausserdem habe ich noch einen Mosquitto Broker auf einen Raspberry Pi installiert. Über diesen Broker soll dann später die ganze Kommunikation ablaufen. Man muss noch den richtigen Port freischalten, damit man ihn benutzen kann. Sonst ist er funktionstüchtig.</p>
24.11	<p>Piotr & Francois: Heute haben wir wieder an den Fahrrädern gearbeitet. Wir mussten beim weissen Fahrrad die hintere Bremse wieder einstellen. Wir haben auch angefangen das Violette Fahrrad zu reparieren. Die Schläuche waren intakt und mussten nicht geflickt werden. Wir mussten einen Teil der hinteren Bremse wieder einhängen. Man muss sie aber noch einstellen. Uns ist aufgefallen, dass man nicht gut Rückwärts treten kann. Das heisst wenn man nicht mehr in die richtige Richtung tritt, versucht das Rad die Kette mitzudrehen. Wahrscheinlich ist also der Freilauf kaputt. Wir müssen uns das mal ein anderes Mal anschauen. (1.5h)</p>
25.11	<p>Francois: Ich habe es geschafft mich mit dem eigenen MQTT Broker zu verbinden. Ich habe es aber nur von meinem Laptop und nicht mit dem GSM-Modul ausprobiert. Dies sollte aber kein Unterschied machen. Die Zugangsdaten zum Broker sind nun folgende: Adresse: hvbg4ioni4q9xr3x.myfritz.net Port: 51883 Ich habe bisher kein Passwort und Benutzername für den Broker aufgesetzt. (0.5h)</p>

27.11	Francois: Ich habe angefangen die finale Schaltung auf einem Breadboard zu stecken. Ich bin aber noch nicht sehr weit gekommen, da ich noch beim Esp32 Pins anlöten muss. (1h)
28.11	Francois: Ich wollte testen, ob wir es schaffen mit einem kleinen Programm in Python die Nachrichten vom Broker empfangen können. Dazu habe ich das Programm fertig geschrieben und so angepasst, damit es mit dem richtigen Broker arbeitet. Ich habe es geschafft Nachrichten zu empfangen. Das heisst auf der Seite des Servers fehlt nur noch, dass man die Daten in eine Datenbank einfügt. (1.5h)
29.11	Francois: Ich habe versucht Pins and den kleinen Esp32 zu löten. Leider hatte ich nur mässig Erfolg. Die angelöteten Drähte sind recht schnell wieder abgefallen. Ich muss mir überlegen wie man das am besten Lötet. Ich bin zufällig auch auf eine Art gestossen, wie man eventuell einen Pin des Esp32 während des Deep-Sleep Hoch halten kann. Es gibt eine Funktion <code>gpio_hold_en()</code> die das vielleicht machen könnte. Man müsste es aber noch testen. (1h)
01.12	Francois: heute habe ich das Programm auf der Seite des Servers geschrieben. Das Programm kann jetzt die Daten vom Broker empfangen und diese in eine SQL-Datenbank einfügen. Es ist gedacht, dass das Programm permanent auf dem gleichen Computer wie der Broker läuft. Ich habe es auch geschafft Drähte mehr oder weniger stabil auf den Esp32 zu löten. Ich möchte es aber mit noch dünneren Drähten versuchen, da ich bisher "normale" Drähte benutzt habe. (3h)
02.12	Francois: Ich habe heute versucht zu messen wie viel Strom die Verschiedenen Teilchen verbrauchen werden. Der Esp32 braucht laut Datenblatt im Deep-Sleep unter einem 1mA. Das GPS braucht so um die 40 mA. Ich konnte das GSM-Modul nicht sehr genau auslesen, da ich von meinem Messgerät beschränkt war. Es kann entweder recht genau aber dann nur bis 300mA messen oder bis zu 10A aber nicht sehr genau. Da das GSM-Modul aber zeitweise über 300mA verbraucht musste ich die ungenaue Option benutzen. Ich habe ca. 10 – 20 mA gemessen, während das GSM-Modul nicht am Machen war. Das heisst insgesamt wird der Tracker im Wartemodus 50 – 60 mA verbrauchen. Batterien haben eine Leistung von ca. 1000 mA. Das heisst das Modul wird im Wartemodus 16 – 20 h halten. Mit dem Senden wird das noch weniger sein. Es lohnt sich also wahrscheinlich nicht die Fahrräder aufzustellen. Wir haben jetzt also folgende Optionen zum Weitermachen: <ul style="list-style-type: none"> • Das Fahrradaufstellen fallen lassen und sich auf die Website konzentrieren • Kommerzielle Tracker zum Aufstellen benutzen. (0.75h)
03.12	Francois: Ich habe heute nochmal nachgerechnet wie viel lange man das GPS Modul mit Alkali-Mangan-Zellen Betreiben kann. Ich habe mit folgendem Stromverbrauch für den Velotracker im Wartemodus gerechnet: <ul style="list-style-type: none"> • Esp32: < 1 mA • Sim800l: 10 – 20 mA • Neo-6m: 40 mA • Insgesamt 50 – 60 mA also 60 mA Der Tracker ist aber nicht die immer im Wartemodus und sendet manchmal was mehr Strom verbraucht. Ich habe deshalb abgeschätzt das dies etwa 1/6 des Verbrauchs ist. Auch muss man beachten, dass man die Batterie nicht komplett entleeren kann. In

	<p>unserem Fall können wir nur 80% der Batterie benutzen. Dies Gibt uns als die Folgende Formel:</p> $(Kapazität * 0.8 / 60) * 5/6$ <p>Eine AAA Alkaline Batterie hat 1200mAh. Das macht $(1200 * 0.8 / 60) * 5/6 = 13.33$ h. Eine AA Alkaline Batterie hat 2700mAh. Das macht $(2700 * 0.8 / 60) * 5/6 = 30$ h. Dies ist nicht genug, um die Fahrräder für eine Längere Zeit aufzustellen.</p>
04.12	<p>Francois: Ich habe heute den Esp32 auf eine Lochrasterplatine gelötet. Ich habe den TX, Pin, RX-Pin, GND-Pin und den VCC-Pin herausgeführt, damit ich mein Programm auf den Esp32 hochladen kann. Als ich den Esp32 aber angeschlossen habe, konnte ich mich nicht mit ihm Verbinden. Ich konnte auch nichts hochladen. Nachdem ich in der Dokumentation nachgeschaut habe, ist mir aufgefallen, dass ich zwei Sachen falsch gemacht habe. Der EN-Pin muss auf Hoch sein damit der Esp32 läuft. Das heisst ich muss ihn mit 3.3V verbinden was ich nicht gemacht habe. Damit man etwas auf den Esp32 hochladen kann muss man den IO0 Pin kurz mit der Masse (GND) verbinden. Nachdem ich diese Sachen gemacht habe, hat der ES32 ein WiFi aufgespannt was heisst, dass er am Laufen ist. Ich konnte aber nicht hochladen. Ich habe dann in einem Forum gelesen, dass man den IO0 Pin mit dem Ground verbinden muss und den EN-Pin kurz auf Tief setzen. Das heisst man müsste vielleicht beim EN-Pin noch einen Schalter löten. (5-6h)</p>
05.12	<p>Francois: Ich habe mit kicad das Schaltbild auf neusten Stand gebracht. Ich habe also die Verbindung zwischen 3.3V (VCC) und EN eingefügt und die zwischen IO0 und dem GND.</p> <p>Das Prozedere, um etwas auf den Esp32 hochzuladen, sieht nach dem derzeitigem Verständnis Folgendermassen aus:</p> <ol style="list-style-type: none"> 1. Der EN-Pin muss auf Hoch sein, wenn er Tief ist, läuft der Esp32 nicht 2. Der IO0 Pin wird mit dem Ground verbunden. Das heisst, dass der Esp32 beim aufstarten nun erwartet, dass etwas hochgeladen wird. 3. Der EN-Pin muss für kurze Zeit auf Tief (GND) und dann wieder auf Hoch (3.3V). Somit wird der Esp32 neugestartet und bootet in den Hochladmodus <p>Der Esp32 wurde vorbereitet und diese Prozedur wurde angewendet. Leider ist es nicht gelungen etwas auf den Esp32 hochzuladen. Im Datenblatt hat es noch ein Schaltbild vom Esp32 welches Kondensatoren benutzt. Man sollte dies noch ausprobieren. (3h)</p>
08.12	<p>Francois: Um das Testen zu verschnellern wird die Elektronik nicht mehr gelötet, sondern auf ein Breadboard gesteckt. So soll man schneller und mit weniger Fehlern Änderungen vornehmen können. Hierzu ist ein kleiner Esp32 mit Füßen versehen worden, damit man ihn auf ein Breadboard stecken kann. (1h)</p>
09.12	<p>Francois: Heute wurde versucht ein Programm auf den Esp32 hochzuladen. Hierzu wurde das empfohlene Schaltbild aus dem Datenblatt benutzt (Seite 24). Die Kondensatoren und der Widerstand mussten mit Beinchen für das Breadboard verlängert werden. Es war trotzdem nicht möglich etwas auf den Esp32 hochzuladen. Es wurde auch eine Spannungsversorgung welche 1 A liefern konnte (empfohlen sind mindestens 500 mA). Dies hat zur Entscheidung geführt, dass man mit dem Entwicklermodul weiterarbeiten soll, obwohl es einen höheren Stromverbrauch hat und grösser ist. (1.5h)</p>
11.12	<p>Francois: Ich habe wieder angefangen die Elektronik auf einem Breadboard aufzubauen. Hierzu wurde aber das Esp32 Entwicklermodul anstatt des kleinen Esp32 verwendet. Der Spannungswandler LM2937 wurde auch auf seine Funktionalität</p>

	<p>getestet. Die Tests waren erfolgreich. Man muss nun noch das GPS- und GSM-Modul anschliessen und testen, ob dies funktioniert. (0.5h)</p>
12.12	<p>Francois: Das System ist in vollem Umfang getestet worden. Die Tests sind am GSM-Modul gescheitert. Nach nachschlagen im Datenblatt ist folgendes aufgefallen: Der Sim800I verträgt 3.4 - 4.4 V empfohlen sind aber 4.0 V. Den Sim800I unter 3.7 V zu betreiben ist nicht empfohlen, da dies die Stabilität beeinträchtigt. Diese Informationen waren schon vorher klar. Dazu gekommen ist aber, dass der Sim800I ab 4.3 V eine Overvoltage Warnung ausgibt und ab 4.4 Volt automatisch ausschaltet. Die Overvoltage Warnungen werden so oft ausgegeben, dass die Bibliothek nicht mehr richtig mit dem GSM-Modul kommunizieren kann. Dies heisst das bisherige Konzept für die Stromversorgung funktioniert nicht mehr. Folgende Konzepte sind getestet worden und haben nicht funktioniert:</p>  <p>Folgende Sachen kann man machen oder muss man noch testen:</p> <ul style="list-style-type: none"> • 2 Dioden seriell zu schalten damit ca. 2.0 V Abfallen, da 4 Alkaline 6 V geben. • Das GSM-Modul separat mit einer Li-Po Batterie Betreiben (man müsste eine Auftreiben) • Mit einem verstellbaren Spannungsregler arbeiten, man müsste aber noch eventuell einen auf die Schnelle kaufen oder besorgen. <p>(5h)</p>
13.12	<p>Francois: Die Idee mit den zwei Dioden funktioniert nicht. Die Spannung schwankt zu stark. Die Spannung ist teilweise zu hoch oder zu tief. Es war möglich eine Li-Po Batterie aufzutreiben. Es handelt sich um eine einzelne Zelle was heisst, dass sie zwischen 4.2 V und 3.7 V hat was die richtige Spannung ist. Die Leistung ist 2500 mAh was wahrscheinlich reicht. Diese muss aber noch aufgeladen werden. Dies ist aber anscheinend ein recht delikater Prozess. Es ist aber gelungen ohne, dass die Batterie beschädigt wurde. Die Batterie hat es geschafft, das GSM-Modul zu betreiben. Anhand des blinken des GSM-Moduls konnte man sehen, dass das Modul wahrscheinlich mit dem Netz verbunden ist. Man soll dies aber überprüfen, ob es wirklich funktioniert.</p>

	<p>Leider hat die Li-Po Batterie nicht geschafft auch den Esp32 und das GPS-Modul mit Strom zu versorgen. Das heisst, dass die beiden von einem anderen Batteriepack versorgt werden. (4h)</p>
15.12	<p>Francois: Nach einigen Versuchen ist es gelungen den Esp32 mit Batterien zu betreiben. Man muss aufpassen welchen GND-Pin man benutzt, da bestimmte nicht sehr gut funktionieren. Auch musste ein Kondensator von 1000 μF neben dem Eingang des Esp32 eingesetzt werden, damit dieser es schafft zu starten.</p> <p>Es ist entschieden was in die Box für den Tracker kommt. Der Esp32 und der Spannungswandler bleiben auf dem Breadboard. Es wird aber alles möglichst flach gemacht. Drüber kommt die Li-Po Batterie, da diese nicht ganz unten in der Box passt. Auf die Li-Po Batterie kommt auf der einen Seite zwei Batteriehalter à 3 Batterien. Auf der anderen Seite sind der Sim800l und der der Neo-6m.</p> <p>Der Esp32 und der Spannungswandler sieht auf dem Breadboard installiert.</p> <p>Man muss sich noch überlegen, wie man die Batterien mit dem Rest verbindet. Man muss die Batterien noch fürs Laden wegnehmen können.</p>
16.12	<p>Der Tracker ist fast fertig. An alle Module sind Kabel gelötet worden. Die Batterien sind auch richtig verkabelt. Der Esp32 und der Spannungswandler sind korrekt auf dem Breadboard installiert. Man muss sich jetzt noch genau überlegen wie die Teile in die Box gesetzt werden. Wahrscheinlich wird zuunterst das Breadboard sein, dann die LiPo-Batterie und darauf die Module und die restlichen Batterien. Das System muss noch in der Box getestet werden.</p>
17.12	<p>Die Anordnung der Komponenten hat sich geändert. Zuunterst kommen die Beiden Module und das Batteriepack, dann die Li-Po Batterie und zuoberst das Breadboard. Für den Gebrauch wird die Box umgedreht, damit das GPS-Modul freie Sicht auf den Himmel hat. Man hat aber immer noch gut Zugang zum Breadboard damit man die Komponenten zusammenstecken kann.</p>
18.12	<p>Der Tracker ist komplett zusammengebaut und getestet worden. Der Tracker hat sich manchmal wegen der TinyGSM Bibliothek aufgehängt. Um dies zu lösen startet sich der Esp32 neu, wenn er zu lange in einem Stadium ist. Der Esp32 ist bei einem Spaziergang zu Fuss mit einem Schlaff Intervall von 20 Sekunden getestet worden. Es hat gut funktioniert. Wie zu erwarten waren die Abstände grösser als 20 Sekunden.</p>