

Projektarbeit

Untersuchungen an Magnetfeldern

Im Rahmen einer Stratosphärenballon-Mission

Sarah Pappert

Matrikelnummer: 4006699

Physik und Technologie für Raumfahrtanwendungen, B.Sc.

Kontakt: sarah.pappert@physik.uni-giessen.de

09.08.2022

WS 2020/21 - SS 2022

Studienprojekt an der Justus-Liebig-Universität Gießen

Arbeitsgruppe Apl. Prof. Dr. Sören Lange

Zweites Physikalisches Institut

Inhaltsverzeichnis

| | | |
|-------|---|----|
| 1 | Einleitung | 3 |
| 2 | Theoretische Betrachtungen | 4 |
| 2.1 | Die magnetische Flussdichte | 4 |
| 2.2 | Magnete und Magnetisierung | 4 |
| 2.3 | Das Erdmagnetfeld | 4 |
| 2.4 | Magnetfelder vermessen | 5 |
| 2.5 | Aufbau der Erdatmosphäre | 6 |
| 2.6 | Der Raspberry Pi als unser „Messgerät“ | 8 |
| 3 | Missionsvorbereitung | 9 |
| 3.1 | Planung der durchzuführenden Experimente | 9 |
| 3.2 | Programmierung und Testmessungen des BME280 | 10 |
| 3.3 | Programmierung und Testmessungen des HMC5883 | 11 |
| 3.4 | Positionsabhängigkeit des Magnetfeldsensors | 13 |
| 3.5 | Entmagnetisierung des Magnetfeldsensors | 14 |
| 3.6 | Weiterführende Bedeutung der Entmagnetisierung | 16 |
| 3.7 | Modularisierung des Codes | 17 |
| 4 | Missionsdurchführung | 18 |
| 4.1 | Aufbau der Sonde | 18 |
| 4.2 | Start des Ballons | 18 |
| 5 | Auswertung des alternativen Experiments | 19 |
| 5.1 | Umwandlung des Magnetfeldsensors in einen Kompass | 19 |
| 5.2 | Auswertung der Messergebnisse | 19 |
| 5.2.1 | Weg zur JLU | 19 |
| 5.2.2 | Weg zur THM und in die Innenstadt | 21 |
| 5.2.3 | Feldweg in Lutherstadt Wittenberg | 22 |
| 5.2.4 | Ampel | 22 |
| 5.2.5 | Bahngleise | 23 |
| 5.2.6 | Telefonleitung | 23 |
| 6 | Fazit | 25 |
| 7 | Anhang | 26 |
| 7.1 | Python Code des Sensors BME280 | 26 |
| 7.2 | Python Code zur Datenauswertung | 28 |
| 7.3 | Python Code des Sensors HMC5883 | 28 |
| 7.4 | Python Code zur Datenauswertung | 30 |
| 7.5 | Modularisierter Code | 30 |
| 8 | Quellenverzeichnis | 33 |

1 Einleitung

Seit Jahrhunderten beschäftigt die Menschen die Frage, was uns umgibt. Schon sehr lange wissen wir von Schallwellen, Licht, kosmischer Strahlung, elektrischen Feldern und natürlich auch von magnetischen Feldern.

Magnetismus stellt sich in erster Linie als Kraftwirkung zwischen Magneten, magnetisierbaren Gegenständen oder bewegten elektrischen Ladungen dar und lässt sich durch das magnetische Feld beschreiben.

Im modernen Industriezeitalter finden wir immer mehr Magnetismus in unserem Leben - schon als Kinder hatten wir magnetisches Spielzeug und haben Kühlschrankmagneten gesammelt, aber auch alle elektrischen Geräte erzeugen ein gewisses Magnetfeld, wenn sie von Strom durchflossen sind.

Voraussetzung für das Leben, wie wir es kennen, ist vor allem auch das natürliche Magnetfeld der Erde. Es hat eine sehr große Bedeutung für uns Menschen, da es uns in erster Linie vor Sonnenwinden schützt, aber auch die entscheidende Rolle für Orientierung und Navigation spielt. Und dennoch ist das Erdmagnetfeld kaum erforscht. Aber warum? Gibt es da nichts spannendes zu erfahren oder liegt es vielleicht eher an dem Messaufwand und werden wir in Zukunft doch noch die ein oder andere Überraschung erleben?

Darüberhinaus hängen auch viele andere Bereiche von Magnetfeldern ab. Hätten Sie gewusst, welche Rolle Magnetismus bei Seeminenten spielt? Oder haben Sie vielleicht eine Drohne schon mal nur im Kreis fliegen sehen? Haben Sie sich schon mal gefragt, wie man magnetisierte Gegenstände auch wieder entmagnetisieren kann? Diese Fragen und noch viele mehr möchte ich im Folgenden beantworten. Wir nehmen Magnetfelder als selbstverständlich hin, dabei sind sie so mannigfaltig und stellen den Ausgangspunkt für faszinierende Forschung dar.

Der Anlass für meine Arbeit ist, dass ich die einmalige Chance bekommen habe, im Rahmen einer Stratosphärenballon-Mission (STRATO) einen Magnetfeldsensor langsam vom Erdboden bis mitten in die Stratosphäre aufsteigen zu lassen und das Erdmagnetfeld zu vermessen.

In meinen Ausführungen beschäftige ich mich aber nicht nur mit Magnetfeldern, sondern vor allem auch mit Messgeräten, um die Magnetfeldstärke zu bestimmen.

Ein besonderer Schwerpunkt liegt auf dem Erdmagnetfeld und dem, was wir bereits darüber wissen und was wir in Zukunft noch herausfinden müssen, aber natürlich auch auf den künstlichen Magnetfeldern, denen wir Menschen im Alltag ausgesetzt sind. Weiterhin möchte ich auf Schwierigkeiten mit den Messgeräten eingehen, mit denen wir auch im Rahmen der STRATO Mission zu kämpfen hatten, dafür unkomplizierte Lösungsansätze aufzeigen und diese mit etwas Theorie zu Magnetfeldern allgemein erklären.

2 Theoretische Betrachtungen

2.1 Die magnetische Flussdichte

Die Flussdichte B ist eine sehr charakteristische Größe für das Magnetfeld, sie kann auf verschiedene Arten berechnet werden:

- über den Fluss Φ durch die Fläche A : $B = \frac{\Phi}{A}$ Gl. 1

- aus der magnetischen Feldstärke H : $B = \mu_r \mu_0 H$ Gl. 2

- mit dem magnetischen Moment M : $B = \frac{M}{\mu_r \mu_0}$ Gl. 3

- über die Lorentzkraft F_L : $B = \frac{F_L}{qv}$ Gl. 4

Die SI-Einheit ist Tesla [$1T$], aber häufig wird auch Gauss [$1G$] = $10^{-5}T$ verwendet. In dieser Arbeit verwende ich die magnetische Flussdichte, um die Stärke von Magnetfeldern darzustellen.

2.2 Magnete und Magnetisierung

Es gibt verschiedene Stoffe, die im Magnetismus eine Rolle spielen. Zum einen sind das natürlich die Permanentmagnete, die wir alle auch aus dem Alltag kennen. Sie werden z.B. bei Verschlüssen eingesetzt, da sie über längere Zeit ihre magnetischen Eigenschaften behalten und vor allem ferromagnetische Stoffe (z.B. Eisen) anziehen. Sie sind dabei auch in der Lage, ferromagnetische Stoffe zu magnetisieren, das bedeutet, dass diese dann vorübergehend auch magnetische Eigenschaften aufweisen, da die Elementarmagnete in ihnen, die vorher beliebig in alle Raumrichtungen ausgerichtet waren, dann spezifisch alle in die selbe Richtung ausgerichtet werden (also wenn man den Nordpol eines Magneten über ein Stück Eisen hält, werden alle Südpole der Elementarmagnete dorthin zeigen, da sich entgegengesetzte Pole immer anziehen) und dadurch baut sich ein Magnetfeld zwischen den Polen auf.

Zu den Magnetfeldlinien sei an der Stelle noch kurz gesagt, dass sie immer durchgängige Linien sind und immer vom Nord- zum Südpol zeigen. Außerdem existieren sie nicht wirklich, sondern sind eher eine gedachte Veranschaulichung.

Zum anderen spielen auch stromdurchflossene Leiter und Spulen eine große Rolle, da jeder Strom ein Magnetfeld (zumindest vorübergehend, wenn sich der Stromfluss zeitlich ändert) erzeugt.

2.3 Das Erdmagnetfeld

Das Erdmagnetfeld durchdringt und umgibt die Erde. Es besteht aus drei Komponenten. Der Hauptanteil des Magnetfelds wird vom Geodynamo im flüssigen äußeren Erdkern hervorgerufen. Ein zweiter Anteil des Erdmagnetfeldes entsteht durch elektrische Ströme in der Ionosphäre und der Magnetosphäre. Der dritte Anteil besteht in dem Feld der remanenten Magnetisierung in Teilen der oberen Erdkruste z. B. Erzlagerstätten.

Durch diese drei Komponenten bedingt, besitzt das Erdmagnetfeld eine gewisse räumliche Variation, also es hat an unterschiedlichen Orten der Erde eine unterschiedliche Stärke. Am Äquator hat das Magnetfeld eine magnetische Flussdichte von ca. $30\mu T$. An den Polen ist der Betrag etwa doppelt so groß. In Mitteleuropa sind es etwa $48\mu T$. [1]

Zusätzlich weist das Erdmagnetfeld eine zeitliche Variation auf, denn im Laufe der Jahre wurde beobachtet, dass die Zone mit einem vergleichsweise schwachen Magnetfeld rund um den

Äquator wächst, während die Regionen mit starkem Magnetfeld an den Polen schrumpfen, wie in Abb. 2 zu sehen.

Der Kompass weist auf weiten Teilen der Erdoberfläche grob in geographische Nord-Richtung. Der Grund dafür ist, dass der geographische Nordpol der Erde eigentlich ihr magnetischer Südpol ist und somit den Nordpol der Kompassnadel anzieht. Das ist die Grundlage für ein wichtiges Mittel zur Orientierung auf der Erde, vor allem wenn keine Sterne o. Ä. zu sehen sind.

Selbst heutzutage ist das Erdmagnetfeld noch nicht sehr detailliert erforscht, es wird zwar mit den Sensoren der GOES-Erdbeobachtungssatelliten der NOAA (National Oceanic and Atmospheric Administration) das Magnetfeld der Erde in einer Höhe von 36000 km gemessen, aber der höhenabhängige Verlauf ist weitestgehend unbekannt.

In dieser Höhe schwankt die magnetische Flussdichte zwischen $50nT$ und $150nT$. [2]

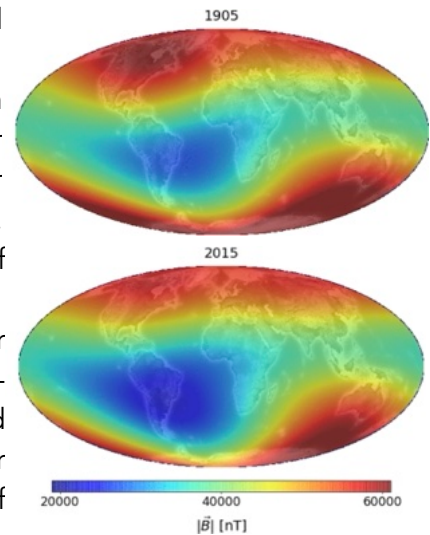


Abb. 2 - räumliche Variation des Erdmagnetfelds: de.wikipedia.org

2.4 Magnetfelder vermessen

Herauszufinden, ob ein Stoff magnetisch oder magnetisiert ist, ist nicht schwer: man muss nur einen Magneten in die Nähe eines potentiell magnetischen Stoffs bringen und testen, ob er angezogen wird. Ist ein Stoff magnetisiert, lässt sich das belegen, wenn er ferromagnetische Stoffe anzieht. Aber über die Stärke des Magneten gibt das nur qualitativ Auskunft - er zieht sehr stark oder nur leicht an.

Es gibt verschiedene Ansätze, um die Stärke eines Magnetfelds exakt zu bestimmen.

Eine ziemlich populäre Möglichkeit, die wir sicherlich schon alle aus der Schule kennen, ist eine **Hallsonde**, welche den Hall-Effekt ausnutzt: Befindet sich ein stromdurchflossener Leiter in einem homogenen Magnetfeld, dann baut sich senkrecht sowohl zur Stromfluss- als auch zur Magnetfeldrichtung über dem Leiter die sogenannte Hall-Spannung auf. Grund dafür ist die auf die Elektronen wirkende Lorentz-Kraft, welche die Elektronen auf eine Seite des Hall-Elements verschiebt und somit für eine Potentialdifferenz sorgt. Die Hallsonde ist vereinfacht gesagt auch ein stromdurchflossener Leiter, der in das zu messende Magnetfeld eingebracht wird und die entstehende Spannung misst. Diese kann über $U_H = R_H \cdot \frac{I_H \cdot B}{d} \rightarrow B = \frac{U_H \cdot d}{R_H \cdot I_H}$ direkt in die

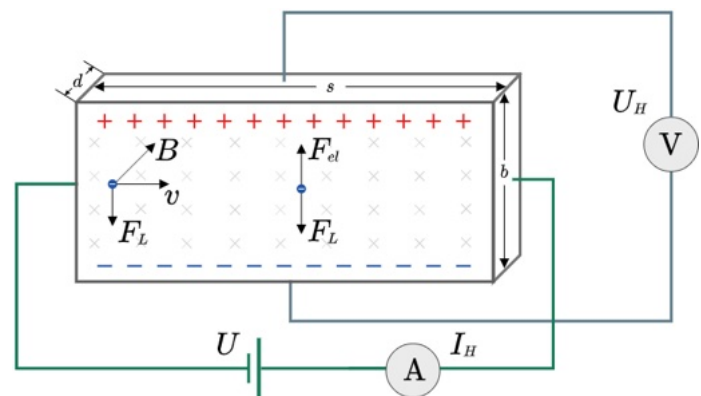


Abb. 3 - Hall-Effekt: physik.gym-wst.de

Flussdichte in Tesla umgerechnet werden. Die benötigten Größen sind in Abb. 3 dargestellt. Darüber hinaus werden aufgrund ihres einfachen Aufbaus auch sehr häufig **XMR-Sensoren** verwendet. Das sind Dünnschicht-Sensoren, die unter Einfluss des magnetischen Flusses direkt ihren Widerstand ändern und daher „X-Magneto-Resistiv“ heißen. Die magneto-resistiven Verfahren unterscheiden sich im Wesentlichen durch das Widerstandsverhalten der verschiedensten Materialien, wenn diese magnetisch durchflutet werden. Es kann sich dabei um ferromagnetische Materialien handeln, um Halbleitermaterialien oder extrem dünne Materialstrukturen, die bei magnetischem Fluss ihre Eigenschaften ändern. [3]

| Magneto-resistives Verfahren | Abk. | Beschreibung |
|---------------------------------|------|---|
| Anisotropic Magneto-Resistive | AMR | Widerstandswert ändert sich durch die Richtung des Magnetfeldes. Legierungen haben eine Materialstruktur bei der der elektrische Widerstand richtungsgebunden, also anisotrop ist. |
| Colossal Magneto-Resistive | CMR | Beim CMR-Verfahren werden Widerstandsänderungen in mikroskopisch dünnen Filme aus Magnesiumoxid genutzt. Änderungsgeschwindigkeiten von ca. 50 μ s. |
| Extraordinary Magneto-Resistive | EMR | Magnetfeldsensoren mit hoher magnetischer Empfindlichkeit gegenüber sehr kurzen magnetisierten Impulsen. Sie bestehen aus Halbleitermaterial und elektrisch leitendem Material. |
| Giant Magneto-Resistive | GMR | Widerstandswert eines metallischen Dünnschichtsystems verändert sich in Abhängigkeit von der magnetischen Feldstärke. Nutzung des Elektronen-Spin-Effekts in ultradünnen Schichten, ändert den Leitungsmechanismus. |
| Tunneling Magneto-Resistive | TMR | Extrem dünne Strukturen. Tunneling von Elektronen durch dünnste isolierende Schichten. Ausrichtung des Magnetfeldes ändert den Widerstandswert. |

Tab. 1 - Eigenschaften der verschiedenen Magneto-resistiven Verfahren: itwissen.info

Bei meinem Sensor HMC5883 handelt es sich um Anisotropic Magneto-Resistive (AMR) Technik. [4] Der AMR-Effekt beruht auf anisotroper (von der Raumrichtung abhängiger) Streuung in ferromagnetischen Metallen. Das heißt, er tritt in Materialien auf, die eine eigene Magnetisierung aufweisen. Stehen der Stromdichtevektor des durch das Material fließenden Stroms und der Magnetfeldvektor der internen Magnetisierung senkrecht aufeinander, so ist der Widerstand des Materials minimal, sind sie parallel zueinander, ist der Widerstand maximal. Der Effekt wird auf eine Verzerrung der Atomorbitale durch die Spin-Ausrichtung im Magnetfeld zurückgeführt. Dadurch ändert sich deren Streuquerschnitt für Leitungselektronen und damit der Widerstand.[5] Diese Sensoren bringen neben geringen Herstellungskosten noch weitere Vorteile mit sich, denn sie zeichnen sich durch eine präzise axiale Empfindlichkeit und Linearität aus. Die Solid-State-Konstruktion dieser Sensoren mit sehr niedriger Querachsenempfindlichkeit ist darauf ausgelegt, sowohl die Richtung als auch die Größe des Erdmagnetfelds von Milli-Gauss bis 8-Gauss zu messen. Deswegen haben wir uns für das STRATO Projekt für diesen Sensor entschieden. Es gibt eine Vielzahl weiterer Teslameter, wie beispielsweise das Faraday-Magnetometer oder Fluxgate Detektoren, die wesentlich aufwendiger in der Herstellung und deswegen für unsere Anwendung nicht relevant sind. Die beiden vorgestellten Methoden sind die gebräuchlichsten.

2.5 Aufbau der Erdatmosphäre

Das Magnetfeld ist auch ein sehr bedeutender Schutz für unsere Atmosphäre, indem es vor Partikelstrahlung (bspw. Sonnenwinde) schützt, diese würden durch ihre hohen Impulse die Gasteilchen aus der Atmosphäre zurück in den Weltraum katapultieren. Nur zusammen machen sie die Existenz von flüssigem Wasser und somit Leben möglich.

Weiterhin bringt unsere Erde eine ausreichend hohe Masse für eine gravitative Bindung der Atmosphäre sowie eine genügend geringe Oberflächentemperatur mit, die keine thermischen Geschwindigkeiten größer der Fluchtgeschwindigkeit verursacht, durch die die Gasteilchen das Schwerefeld der Erde wieder verlassen könnten.

Unsere Atmosphäre besteht hauptsächlich aus Stickstoff (78.08 %_{vol}) und Sauerstoff (20.95 %_{vol}) sowie einigen Edelgasen, wie bspw. Argon (0.93 %_{vol}). Zudem sind noch weitere Verbindungen in sehr geringen Anteilen vorhanden, wie beispielsweise Kohlenstoffdioxid und Ozon.[6]

Ein weiterer bedeutender Bestandteil ist Wasserdampf, sein Anteil liegt bei durchschnittlich 2.06 %_{vol}, variiert allerdings stark nach Ort und Wetterlage.

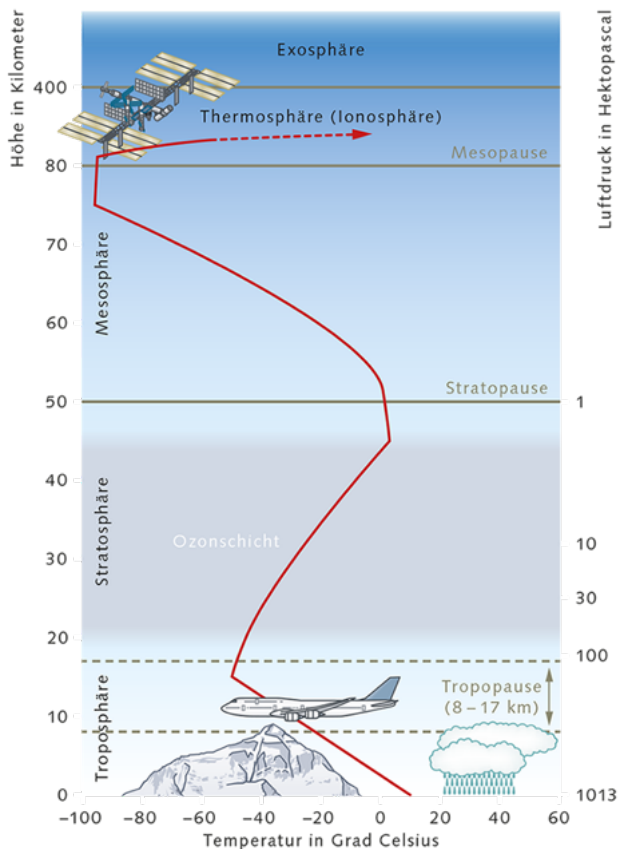


Abb. 4 - Schichten der Erdatmosphäre: worldoceanreview.com

Weiterhin lässt sich die Atmosphäre in verschiedene Schichten einteilen - anhand des Temperaturverlaufs in Troposphäre, Stratosphäre, Mesosphäre, Thermosphäre und Mesosphäre.

Da unser Ballon voraussichtlich 35km hoch steigen wird, sind für uns vor allem die Troposphäre sowie die Stratosphäre interessant.

Die **Troposphäre** schließt sich direkt an den Erdboden an. In ihr spielt sich der Großteil des Wetters ab. Der Hauptanteil der Sonnenenergie wird am Erdboden umgesetzt und in die Atmosphäre abgegeben, weswegen diese in Bodennähe am wärmsten ist. Sie wird also nicht direkt durch Sonneneinstrahlung erwärmt, mit zunehmender Höhe kann man somit einen deutlichen Temperaturabfall erkennen.[7]

Ihre Dicke variiert zwischen 8km an den Polen und 17km am Äquator.

Die **Stratosphäre** folgt nach der Tropopause. Sie erstreckt sich bis zur Stratopause in ungefähr 50km Höhe. Im Vergleich zur Troposphäre liegt hier ein inverser, also steigender Temperaturverlauf vor, der hauptsächlich durch das in der Stratosphäre befindliche Ozon verursacht, das UV-

Strahlung aus dem Sonnenlicht absorbiert und dabei elektromagnetische Strahlung in Wärme umwandelt. Infolgedessen steigt die Temperatur von ca. -60°C bis auf knapp unter 0°C an.

Bereits 1934 stieg der bemannte russische Stratosphärenballon CCCP-1 (Союз Советских Социалистических Республик Один) auf über 19.000 Meter, also bis in die Stratosphäre hinein. Mittlerweile sind auf den meisten Teilen der Welt täglich mehrere Wetterballons in der Stratosphäre unterwegs.

In Deutschland gibt es aktuell 16 registrierte Standorte für regelmäßige Wetterbeobachtung mithilfe von Stratosphärenballons. Die Ballons starten von dort aus mindestens 2-4 mal täglich, bei Bedarf sogar stündlich.[8] Viele Informationen aus der Höhe werden mittlerweile aus den Daten von Flugzeugen und Satelliten ermittelt, aber in der vertikalen Auflösung von Wind, Temperatur und Feuchte sind die Radiosonden der Wetterballons weiterhin unschlagbar und gelten somit immer noch als wichtige Referenz bei der Wettervorhersage und darüberhinaus sogar bei der Kalibrierung von weiteren Fernerkundungsdaten wie z. B. Satellitendaten.[9]

Nach dem radio-physikalischen Zustand der Atmosphäre lässt sie sich weiterhin in Neutrosphäre (Gase überwiegend im neutralen, also nicht-ionisierten Zustand), Ionosphäre (ionisierte Gase, in der Thermosphäre eingelagert, >80 km), Plasmasphäre (vollständige Ionisation aller Teilchen, >1000 km) und Magnetosphäre unterteilen.

Als **Magnetosphäre** bezeichnet man das Raumgebiet um ein astronomisches Objekt, in dem geladene Teilchen von dessen Magnetfeld beeinflusst werden. Die äußere Begrenzung wird Magnetopause genannt und die Ionosphäre bildet die innere Begrenzung zur neutralen Atmosphäre.

Unsere Magnetosphäre wird vor allem durch das vom Sonnenwind getragene Magnetfeld geformt. Der Sonnenwind staucht die Magnetosphäre auf der Sonnenseite zusammen und zieht sie auf der Nachtseite zu einem Magnetschweif auseinander (siehe Abb. 5). Die Form der

Magnetopause ist somit nicht statisch, sondern ändert sich zeitlich sehr stark in Abhängigkeit vom Impuls des Sonnenwinds.[10]

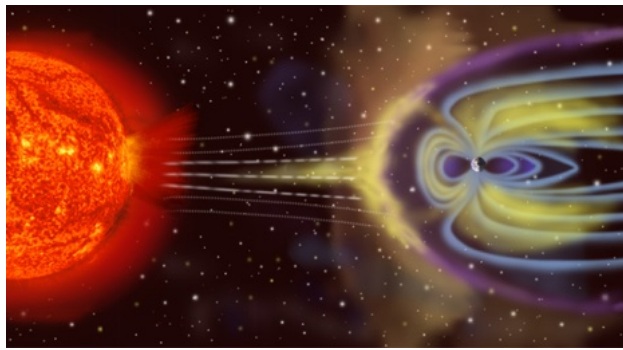


Abb. 5 - Magnetosphäre: de.wikipedia.org

2.6 Der Raspberry Pi als unser „Messgerät“

Es gibt eine Reihe von sogenannten „Mikrocontrollern“, also Ein-Chip-Computersystemen, mit prominenten Beispielen wie Arduino, ESP und Raspberry Pi. Dabei gibt es ein grundlegendes Unterscheidungsmerkmal: Der Arduino (und genau so der ESP als eine Art modernere Version) haben kein eigenes Betriebssystem, man installiert am PC eine Entwicklungsumgebung, über die man die Technik implementiert.

Der Raspberry Pi hingegen ist ein vollständiger Computer, als Betriebssystem kommt häufig eine angepasste Linux-Distribution (Raspberry Pi OS) mit grafischer Benutzeroberfläche zum Einsatz. Deswegen ist es sehr komfortabel, beispielsweise Sensoren am Raspberry Pi zu programmieren. Da wir im Rahmen unseres Projekts mit mehreren Sensoren arbeiten werden, haben wir uns für einen Raspberry Pi und die Programmiersprache Python als Grundlage entschieden. Daher bietet es sich an, den Aufbau des Mikrocontrollers kurz anzuschauen, damit alle folgenden technischen Aspekte gut nachvollziehbar sind.

Es gibt eine Vielfalt an Modellen, für den Einsatz im STRATO Ballon haben wir uns für einen Pi Zero entschieden, da er der kleinste und leichteste Pi ist und wir etwas auf das Gesamtgewicht achten müssen. Für die Tests der Sensoren kam beispielsweise auch ein Pi 3 Model B+ zum Einsatz, welcher etwas größer ist, aber im groben Aufbau gleichen sich alle Raspberry Pi's:

Das Kernelement ist eine Reihe von Pins (GPIOs) auf der Oberseite (Abb. 6a), an die in unserem Fall die Sensoren angeschlossen werden. Darüberhinaus verfügen sie noch über eine Reihe weiterer Anschlüsse, in erster Linie Mini-USB für die Stromzufuhr und in unserem Fall noch USB A, Ethernet, A/V und HDMI. Zudem befindet an der Unterseite (Abb. 6b) ein Mikro-SD-Karten-Slot. Auf dieser SD-Karte wird dann das entsprechende Betriebssystem installiert und dorthin speichert der Pi auch alle Daten.



Abb. 6a - Vorderseite des Raspberry Pi 3 B+ : Sarah Pappert

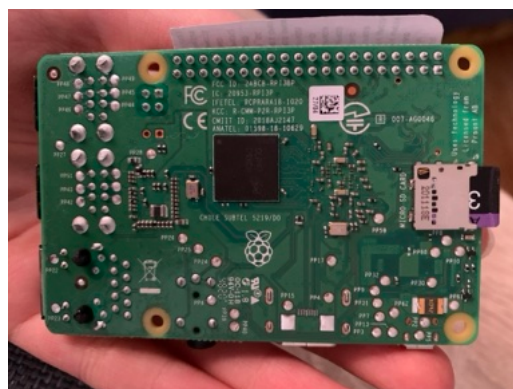


Abb. 6b - Rückseite des Raspberry Pi 3 B+ : Sarah Pappert

3 Missionsvorbereitung

3.1 Planung der durchzuführenden Experimente

Als feststand, dass ein weiterer Start eines Stratosphärenballons an unserer Uni stattfinden wird, mussten natürlich als erstes die Experimente geplant werden, die während des Flugs durchgeführt werden sollen. Wir haben uns einerseits für ein Mikrobio-Experiment entschieden, was Bakterien und Sporen unter den Bedingungen der höheren Atmosphärenschichten untersuchen soll, also beispielsweise den Einfluss der kosmischen Strahlung, den höheren UV-Anteil im Licht, höhere Ozonkonzentration und so weiter. Dafür wird sich eine Probe zusammen mit Sensoren für die relevanten Umwelteinflüsse im Ballon befinden. Zudem wird eine Vergleichskultur in einem Klimaschrank der Uni beobachtet.

Darüberhinaus beschäftigen wir uns auch allgemein mit der Programmierung von Sensoren und der Auswertung der gemessenen Daten.

Einer dieser Sensoren ist der **Magnetfeldsensor** HMC5883. Die Funktionsweise wurde im Abschnitt 2.4 bereits kurz erläutert.

Die Idee für die Messung ist, dass das Erdmagnetfeld in Abhängigkeit von der Höhe über dem Erdboden seine Stärke ändert, wie jedes andere Magnetfeld auch, das in zunehmender Entfernung schwächer wird. Es gilt Allgemein $B, H \propto \frac{1}{r}$, wobei r die für Entfernung steht.

Allerdings wissen wir, dass das Erdmagnetfeld weder zeitlich noch räumlich konstant ist, deswegen stellt sich die Frage, was für einen konkreten Zusammenhang wir in Abhängigkeit von der Höhe finden. Gibt es eventuell Schichten in der Atmosphäre, in denen das Magnetfeld punktuell stärker oder schwächer ist?

Des Weiteren müsste sich die Gestalt der Magnetosphäre bei so einer Messung erkennen lassen: tagsüber (also auf der Sonnenseite) müsste das Magnetfeld viel früher und stärker abnehmen als nachts (vgl. Abb. 5). Da unserer Flug tagsüber stattfinden soll, erhoffen wir uns, bereits ein paar Kilometer oberhalb der Erdoberfläche kleine Änderungen feststellen zu können.

Hinzukommt, dass sämtliche technischen Geräte ein Magnetfeld erzeugen, da man sie als „stromdurchflossene Leiter“ betrachten kann und somit annähernd gilt: $B = \frac{\mu I}{2\pi r}$, wobei I der fließende Strom und r der Abstand ist. Deswegen nehmen wir an, dass wir auf der Erdoberfläche sowieso ein stärkeres Magnetfeld messen, aber mit zunehmender Höhe sollte deren Einfluss merklich abnehmen.

Ein anderer Sensor ist der BME280 für **Lufttemperatur, Luftdruck und Luftfeuchtigkeit**. Man muss dazu sagen, dass sich im Payload des Ballons sowieso ein Datalogger befinden wird, der beispielsweise Innen- und Außentemperatur aufzeichnen wird, dieser Sensor dient mehr oder weniger zum Vergleich.

Wir wissen, dass sich diese Werte im Laufe des Flugs ändern werden. Mithilfe bestehender Modelle der Atmosphäre kann man aus Druck und Temperatur leicht Rückschlüsse auf die Steighöhe ziehen. Wir möchten prüfen, ob unsere Daten den gleichen Verlauf, wie in Abb. 4 dargestellt, aufweisen werden.

Aufgrund der bereits vorgestellten Vorteile haben wir uns entschieden, alle Sensoren über Raspberry Pi's zu programmieren. Dadurch, dass sich die MuonPi Gruppe uns angeschlossen hat mit einem Experiment zur Detektion kosmischer Schauer, können wir auch sehr von ihrem Know-how über die Technik (insbesondere Raspberry Pi) profitieren.

3.2 Programmierung und Testmessungen des BME280

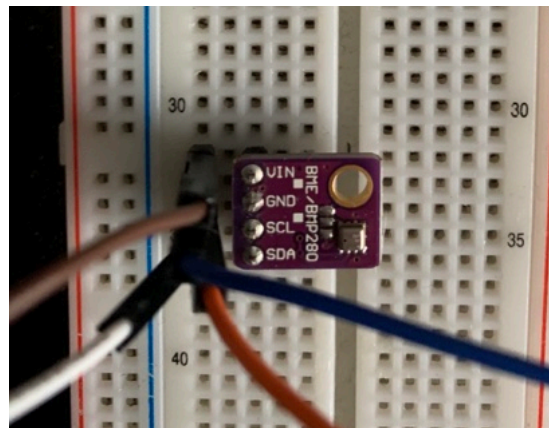


Abb. 7 - BME280: Sarah Pappert

Der BME280 ist ein sehr gebräuchlicher Sensor, tatsächlich wurde er auch beim letzten Strato-Start verwendet. Er war sehr leicht zu programmieren und lieferte direkt realistische Messwerte.

Es handelt sich hierbei um einen klassischen I²C Sensor, somit wird die Spannungsversorgung V_{in} des Sensors an die 3.3V des Raspberry Pis angeschlossen, SCL an SCL (GPIO 3) und SDA an SDA (GPIO 2) und GND an GND. Um den Sensor zu programmieren, muss als erstes das Modul `smbus` importiert werden. Ein SMBus ist ein Zweileiterbus, der von den I²C Sensoren genutzt wird, dieses Modul stellt dementsprechend die I²C-Tools zur Verfügung. Das Programm vollzieht folgende Schritte: Zunächst muss der Sensor initialisiert werden, die Chip-ID muss gelesen und verifiziert werden. Mit der Funktion `readBME280A11` werden dann tatsächlich die Sensordaten ausgelesen, wobei Druck und Temperatur aus je drei Registern und die Feuchtigkeit aus zwei Registern gelesen wird. Die Funktion besorgt auch gleich noch alle Umrechnungen, sodass sie bereits die finalen Werte für Druck, Temperatur und Feuchtigkeit zurückgibt.

Hinzu kommt die `append_new_line` Funktion, welcher man einen Dateinamen und Daten übergeben muss. Sie prüft zunächst, ob die genannte Datei bereits vorhanden ist und hängt dann eine Zeile an sie an. Ist sie nicht vorhanden, wird eine neue Datei erstellt und unsere Messwerte werden als String hineingeschrieben. So werden unsere Daten gesichert und direkt in einer .txt Datei mit Trennzeichen ablegt, sodass man sie gut einlesen und plotten kann.

Die Codes sind im Anhang und die für das Verständnis wichtigen Stellen sind **fett** gedruckt.

Um die Empfindlichkeit des Sensors zu testen, hab ich ihn vor ein offenes Fenster (im Februar) und vor die warme Heizung gestellt und er hat sofort deutliche Veränderungen der Lufttemperatur angezeigt. Um die Luftfeuchtigkeit zu erhöhen, hab ich mit dem Wasserkocher ein bisschen Wasserdampf im Raum verteilt, auch das hat der Sensor erkannt. Zum Variieren des Luftdrucks könnte man in einem luftdichten Gefäß (Plastiktüte) mithilfe einer Pumpe ein Vakuum oder höheren Druck erzeugen, das war mit meinen technischen Möglichkeiten leider nicht möglich.

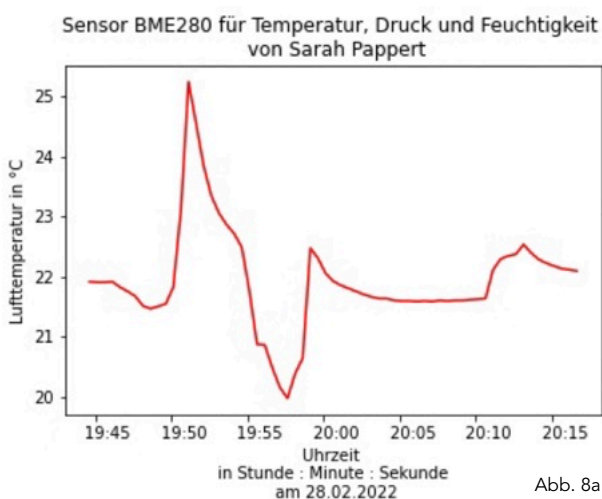


Abb. 8a

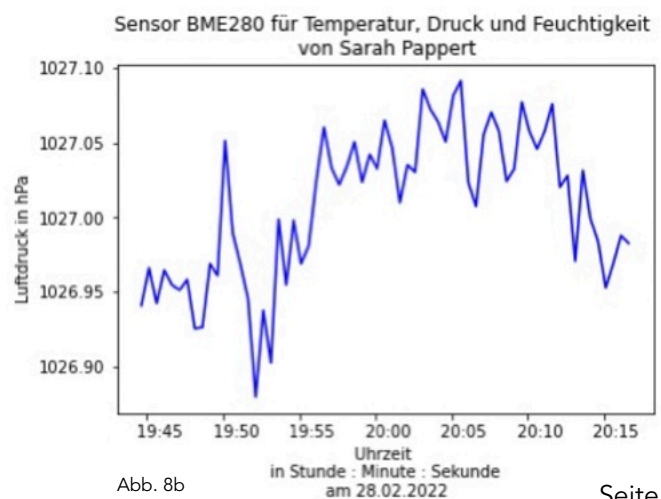


Abb. 8b

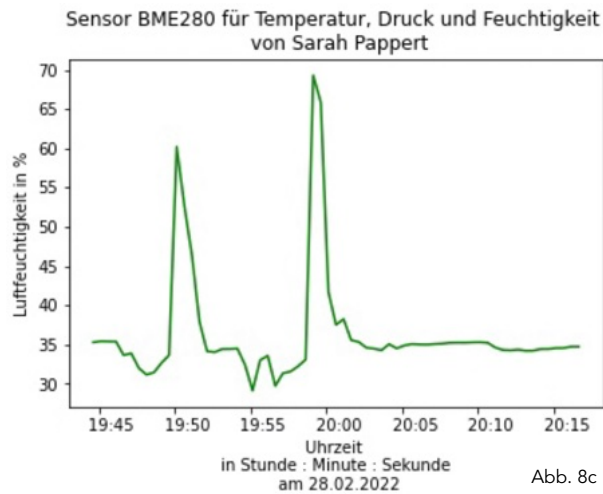


Abb. 8 - Messwerte Probemesseung BME280: Sarah Pappert (mit Python)
a) Temperaturdaten, b) Druckdaten, c) Feuchtigkeitsdaten

3.3 Programmierung und Testmessungen des HMC5883

Das Board HMC5883 des Sensors GY271 ist eigentlich dafür gedacht, die Richtung eines Magnetfelds zu bestimmen und wird meistens als Kompass eingesetzt, da es seinen Widerstand in Abhängigkeit der Richtung des äußeren Magnetfelds ändert. Dementsprechend war es etwas schwierig, an die eigentliche Stärke des Magnetfelds heranzukommen. Da es sich ebenfalls um einen I²C Sensor handelt, sind die Anschlüsse absolut äquivalent zum BME280. Den Pin `DRDY` für schnelles Auslesen nutzen wir nicht.

Im Hauptprogramm `main()` muss man den Sensor erstmal initialisieren, damit er weiß, in welcher Frequenz, Größenordnung (Fieldrange) und in welchem Modus er messen soll. Ich habe mich mithilfe des Datenblatts für 15Hz, 2.5 Gauss und eine kontinuierliche Messung entschieden.

| GN2 | GN1 | GN0 | Recommended Sensor Field Range | Gain (LSb/Gauss) | Digital Resolution (mG/LSb) | Output Range |
|-----|-----|-----|--------------------------------|------------------|-----------------------------|----------------------------|
| 0 | 0 | 0 | ± 0.88 Ga | 1370 | 0.73 | 0xF800–0x07FF (-2048–2047) |
| 0 | 0 | 1 | ± 1.3 Ga | 1090 (default) | 0.92 | 0xF800–0x07FF (-2048–2047) |
| 0 | 1 | 0 | ± 1.9 Ga | 820 | 1.22 | 0xF800–0x07FF (-2048–2047) |
| 0 | 1 | 1 | ± 2.5 Ga | 660 | 1.52 | 0xF800–0x07FF (-2048–2047) |
| 1 | 0 | 0 | ± 4.0 Ga | 440 | 2.27 | 0xF800–0x07FF (-2048–2047) |
| 1 | 0 | 1 | ± 4.7 Ga | 390 | 2.56 | 0xF800–0x07FF (-2048–2047) |
| 1 | 1 | 0 | ± 5.6 Ga | 330 | 3.03 | 0xF800–0x07FF (-2048–2047) |
| 1 | 1 | 1 | ± 8.1 Ga | 230 | 4.35 | 0xF800–0x07FF (-2048–2047) |

Tab. 2 - Gain Settings: digikey.de

Die Sensordaten werden mit dem Kommando `bus.read_i2c_block_data` gelesen, weil man die Standardfunktionen des importierten Moduls `smbus` verwenden kann. Nun konvertiere ich sie noch, indem ich sie von einem Einer- in ein Zweierkomplement umwandle und dadurch mit einem Vorzeichen behafte. Dafür sorgen die Funktionen `twos_complement` und `convert_sw`.

Jetzt muss nur noch entsprechend des Datenblatts der zur **Fieldrange** passende **Gainfaktor** angewandt werden, damit wir vernünftige Werte in der Einheit mG erhalten. In unserem Fall also $1.52 \frac{mG}{LSb}$. Daraufhin folgt wieder die `append_new_line` Funktion zur Datenspeicherung.

Beim Testen dieses Sensors ist mir noch eine Besonderheit aufgefallen, die sich nun auch im Programmcode widerspiegelt: wenn irgendwie die Verbindung zwischen Sensor und Raspberry Pi getrennt wird (z.B. loses Kabel), wird etwas in die .txt Datei geschrieben, wodurch sie nicht mehr zu öffnen ist. Da beim Flug des Ballons mit starken Winden und einem unsanften Aufprall zu rechnen ist, haben wir uns entschieden, in regelmäßigen Abständen in neue Dateien zu schreiben, damit im Fall der Fälle nicht alle Messwerte verloren gehen würden. Der entscheidende Programmteil beginnt mit einer Endlosschleife `while 1:` und einem Dateinamen, der die momentane Uhrzeit enthält und innerhalb dieser Schleife immer wieder erstellt wird. Daraufhin folgt eine endliche for-Schleife `for i in range (0, 30) :`, die in zehnstündigen Abständen (durch `sleep(10)`) 30 Mal durchlaufen wird, also fünf Minuten lang. Innerhalb dieser Schleife werden die Messwerte in die Datei geschrieben. Solange die Datei den gleichen Namen hat, werden die Messwerte einfach weiter dazu geschrieben. Nach den fünf Minuten wird wieder die äußere Endlosschleife durchlaufen, also ein Dateiname mit der momentanen Uhrzeit erstellt. Wenn es nun wieder in die for-Schleife geht, hat die Datei, in die geschrieben werden soll, einen anderen Namen und wird somit alle fünf Minuten neu angelegt und enthält jeweils 30 Datensätze. Dabei wird die „alte“ Datei immer automatisch geschlossen und gesichert.

Bei den ersten Testmessungen zeigte sich aber auch noch ein weiteres Problem: die Messwerte waren absolut sinnlos, aber immerhin stabil ($\approx 130 \cdot 10^{-4} T$). Um zu prüfen, ob sie sich überhaupt ändern und wenn ja, wie empfindlich der Sensor ist, habe ich einen schwachen Kühlschrankmagneten ($\approx 5 \cdot 10^{-2} T$) in seine Nähe gebracht. Es hat sich auch eine deutliche Veränderung gezeigt, aber nach wie vor in der falschen Größenordnung. Um den realistischen $\approx 40 \cdot 10^{-6} T$ des Erdmagnetfelds näher zu kommen, habe ich die Messung erstmal nach draußen verlagert, um dem Einfluss aller elektrischen Geräte zu entgehen.

Eine grobe Zusammenrechnung, wie stark sie den Sensor beeinflussen könnten:

| | |
|--------------------------|---------------------------|
| Erdmagnetfeld in Europa | Ca. $4 \cdot 10^{-5} T$ |
| direkt vor dem Netzteil | Ca. $1.5 \cdot 10^{-3} T$ |
| 20cm Abstand vom Router | Ca. $7 \cdot 10^{-8} T$ |
| Hochspannungsleitung | Ca. $2 \cdot 10^{-5} T$ |
| 3 Handys | Ca. $3 \cdot 10^{-6} T$ |
| 15cm Abstand vom PC | Ca. $1 \cdot 10^{-5} T$ |
| 15cm Abstand vom Drucker | Ca. $2 \cdot 10^{-6} T$ |
| Gesamt | Ca. $1.6 \cdot 10^{-3} T$ |

Tab. 3 - Magnetfelder von Haushaltsgeräten: emf-portal.org, ptb.de

Das ist schon nur noch eine Zehnerpotenz kleiner als der Messwert, also gar nicht mal so unrealistisch. Aber auch draußen änderte sich der Wert leider kaum. Selbstverständlich kommt noch hinzu, dass auch der Raspberry Pi selbst den Sensor beeinflusst.

Die erste Lösungsstrategie war, die Programmierung zu ändern. Ich hab die **Fieldrange** von **8.1Ga** auf **2.5Ga** geändert und den entsprechenden **Gainfaktor** berücksichtigt, den ich vorher

außen vor gelassen habe. Nun erhalten wir immerhin schon mal Werte in mG , also in der Größenordnung $9 - 200 \mu T$, innerhalb der sich auch das tatsächliche Erdmagnetfeld befindet. Jetzt gilt es, eine Strategie zu finden, damit wir nur noch zwischen $40 - 50 \mu T$ liegen.

3.4 Positionsabhängigkeit des Magnetfeldsensors

Wie in den Abschnitten 2.3 und 3.3 bereits erwähnt, ändert der AMR-Sensor seinen Widerstand in Abhängigkeit der Richtung und Stärke des äußeren Magnetfelds. Das bedeutet für uns, dass der Sensor nicht in allen Ausrichtungen dieselben Messwerte liefert, also es macht einen Unterschied, ob er beispielsweise flach auf dem Tisch liegt oder senkrecht steht. Deswegen war mein erster Ansatz, um realistischere Messwerte zu bekommen, die Entfernung zwischen Sensor und Raspberry Pi (bzw. auch weitere elektrische Geräte) durch längere Kabel zu vergrößern und dann die Ausrichtung des Sensors zu verändern. In Abb. 9 sieht man die verschiedenen Positionen und in Abb. 10 den entsprechenden Plot der Messwerte. Das Optimum ergab sich für diesen Sensor, wenn er parallel zur Erdoberfläche positioniert wird, allerdings gibt es dabei auch Unterschiede zwischen den Sensoren, es muss also für jeden individuell ausgetestet werden. Ich habe es mit drei baugleichen Sensoren getestet und drei leicht unterschiedliche Ergebnisse bekommen, möchte aber nur einen Sensor exemplarisch darstellen:

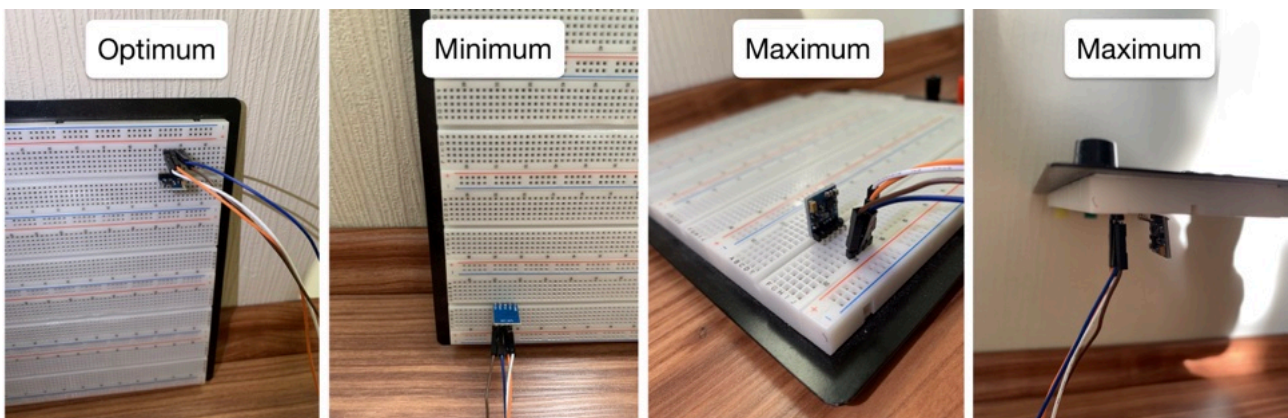


Abb. 9 - Positionierung HMC5883: Sarah Pappert

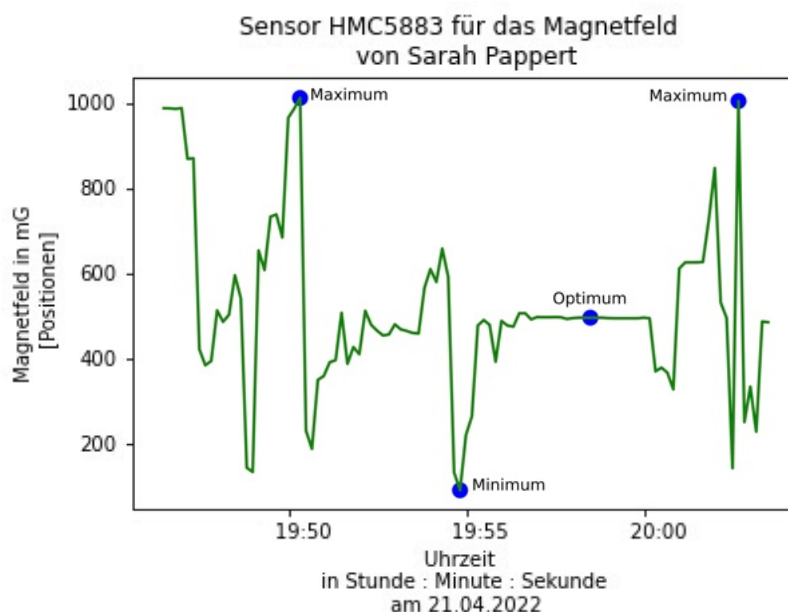


Abb. 10 - Messwerte Positionierung HMC5883: Sarah Pappert (mit Python)

In der meiner Meinung nach idealen Positionierung habe ich den Sensor dann nochmal draußen auf der Wiese getestet und recht stabile Werte rund um $\approx 48\mu T$ erhalten:

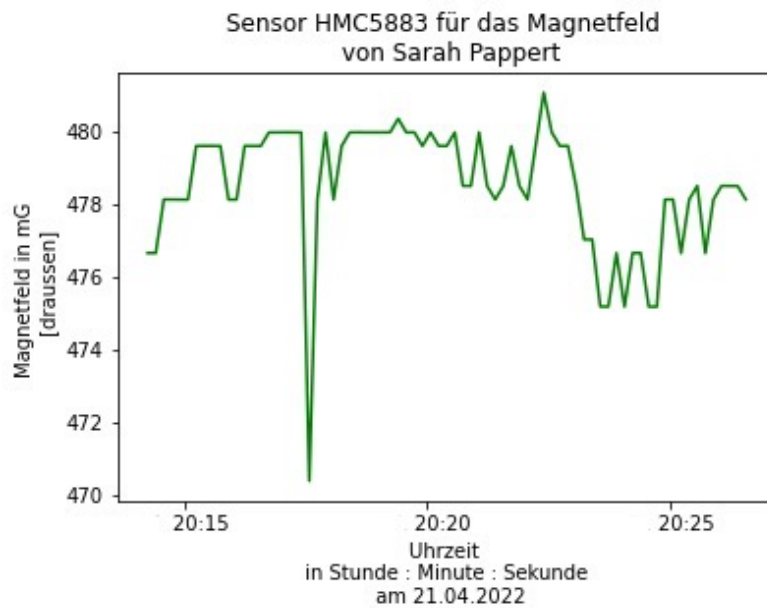


Abb. 11 - Messwerte HMC5883 draußen: Sarah Pappert (mit Python)

3.5 Entmagnetisierung des Magnetfeldsensors

Auch wenn die Werte jetzt schon akzeptabel sind, habe ich noch etwas weiter recherchiert und dabei herausgefunden, dass dieser Magnetfeldsensor auch häufig in Drohnen eingesetzt wird. Ich habe also Holger Steinhaus vom Flugplatz in Lüsse kontaktiert und von ihm erfahren, dass der Sensor in den Drohnen zur Navigation eingesetzt wird und tatsächlich oft Probleme verursacht. Bei unsachgemäßem Transport oder eigenständig durchgeführten Reperaturen (bspw. Schweißen mit Starkstrom und somit starkem Magnetfeld) wird der Sensor selbst magnetisiert, das „innere“ Magnetfeld kommt somit durcheinander, dementsprechend kann auch das äußere Magnetfeld nicht mehr vernünftig gemessen werden und man erhält viel zu hohe Werte, bzw. bei den Drohnen ist der Richtungsvektor leicht schief und somit kann der Kurs nicht mehr korrigiert werden und sie fliegt nur noch Kreise.

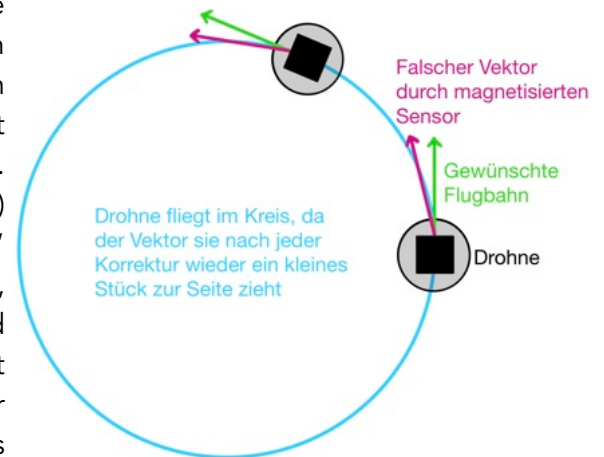


Abb. 12 - Magnetisierte Drohne: Sarah Pappert

Ich habe meinen Sensor wahrscheinlich magnetisiert, als ich den Kühlschrankmagnet für die Probemessung in seine Nähe gebracht habe. Er war zwar nur sehr schwach, aber der Sensor ist auf die Stärke des Erdmagnetfelds ausgelegt und hat somit ein paar Zehnerpotenzen zu viel abbekommen. Glücklicherweise ist eine in meinem Fall leichte Magnetisierung kein irreparabler Totalschaden. Was magnetisiert werden kann, kann in gewissem Maße auch wieder entmagnetisiert werden. Dafür habe ich ein paar **Verfahren** getestet:

Ziel der Entmagnetisierung ist, dass sich die Elementarmagnete im Material wieder willkürlich und ungeordnet ausrichten. Eine Möglichkeit dafür ist ein **Aufprall auf eine harte Oberfläche** und somit eine starke Erschütterung im Material.[11] Dafür hab ich Schraubenzieher zunächst an einen

Magneten gehalten, wodurch Schrauben an ihm haften geblieben sind. Dann habe ich ihn auf einen Steinboden fallen lassen und die Schrauben haften danach kaum noch, also war er weniger stark magnetisch.

Außerdem können die Elementarmagnete durch **starke Hitze** in ihren ungeordneten Ausgangszustand zurückfinden. Im Idealfall würde man dafür die Flamme eines Bunsenbrenners nutzen, ich hatte allerdings gerade nur ein Teelicht zur Verfügung und selbst damit hat die Entmagnetisierung geklappt, wie in Abb. 13 a) zu sehen ist.

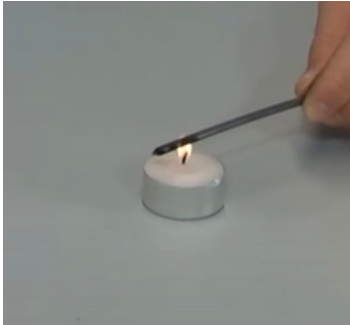


Abb. 13a - Erhitzen zum Entmagnetisieren: Sarah Pappert



Abb. 13b - Entmagnetisierwerkzeug: Sarah Pappert

Eine weitere Möglichkeit ist, mit einer Spule ein **Wechselfeld** zu erzeugen. Diese Technik wird auch in kleinen Magnetisier-/Entmagnetisier-Werkzeugen (Abb. 13 b) angewandt. Ich habe mal eins davon getestet und eine definitiv nicht magnetische Nadel zuerst magnetisiert (sie blieb am entmagnetisierten Schraubenzieher hängen) und dann wieder entmagnetisiert (sie blieb nicht mehr hängen).

Man kann selbst ein Wechselfeld ohne Spule erzeugen, indem man einen leicht magnetisierten Schraubenzieher über das Werkzeug pendeln lässt. Diese Methode fand ich am passendsten für den Magnetfeldsensor.



Abb. 13c - Wechselfeld mit Schraubenzieher zum Entmagnetisieren: Sarah Pappert

Nachdem ich die im Abschnitt 3.4 beschriebene Messung draußen durchgeführt habe, habe ich die Entmagnetisierung durchgeführt mit unglaublichen Ergebnissen: der Sensor liefert nun tatsächlich $\approx 5\mu T$ weniger, da sich die Messwerte nun rund um $\approx 43\mu T$ bewegen. Ich bin äußerst zufrieden mit diesem Ergebnis und bin überzeugt, dass der Sensor nun auch wirklich realistische Werte liefert und wir damit tatsächlich Messungen am Erdmagnetfeld durchführen können.

Allerdings kann während des Ballonflugs nicht gewährleistet werden, dass sich der Sensor wirklich in der richtigen Position befindet und seine Position auch nicht ändert. Problematisch könnte es werden, wenn während des Fluges etwas verrutscht, das würden wir als Ausreißer bzw. Stufe im Plot unserer Messwerte erkennen. Das Ergebnis würde ebenfalls verfälscht werden, wenn das Payload sich beim Auf- oder Abstieg um eine vertikale Achse dreht. Wir hätten somit ständige Schwankungen, also eine Art Zickzacklinie. Allerdings müsste insgesamt dennoch ein Trend erkennbar sein, wenn man dieses Zickzack anfährt, da das Erdmagnetfeld beim Aufstieg langsam schwächer und beim Abstieg schnell stärker wird.

In Abb. 12 c) sieht man, wie ich den Schraubenzieher über dem Sensor pendeln lasse. Die Position ist dabei absolut unverändert zur vorherigen Messung geblieben. Die Messwerte sind in Abb. 13 dargestellt:

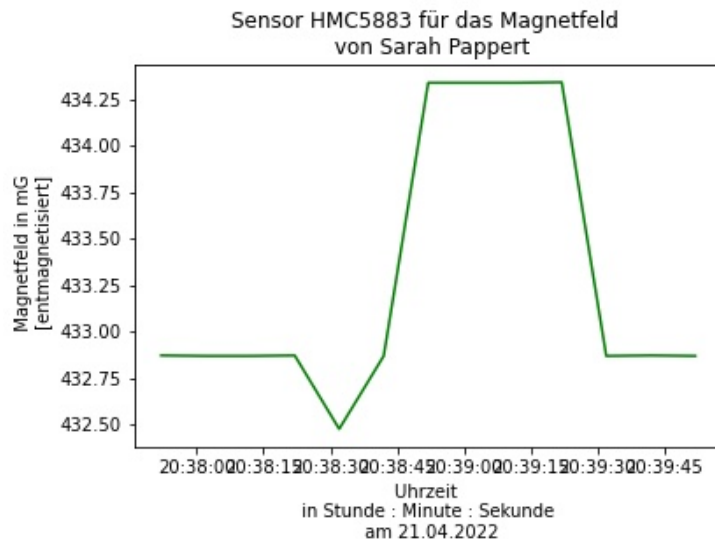


Abb. 14 - Messwerte HMC5883 entmagnetisiert: Sarah Pappert (mit Python)

3.6 Weiterführende Bedeutung der Entmagnetisierung

Magnetisieren und Entmagnetisieren spielt logischerweise bei Werkzeugen eine sehr große Rolle - manchmal ist es vorteilhaft, wenn die Schrauben am Schraubenzieher haften, manchmal müssen Magnetfelder aber auch von den Bauteilen ferngehalten werden, um beispielsweise eine Magnetisierung von Orientierungshilfen (Kompass und Co.) zu vermeiden, sensible Bauteile vor Schäden zu schützen (z.B. SMD-Bauteile sollten nicht an magnetischen Pinzetten haften bleiben) oder um automatische Zielerfassungssysteme zu umgehen. Eine besonders große Rolle spielt dies bei U-Booten und Schiffen. Während der Bauzeit des Schiffes entsteht eine starke Permanentmagnetisierung, abhängig von den verwendeten Materialien (z.B. stark magnetisches Eisen oder nichtmagnetische Glasfaser und Aluminiumlegierungen) und Verfahren (z.B. Schweißen mit Starkstrom und somit starkem Magnetfeld oder vorwiegend Nietverbindungen).

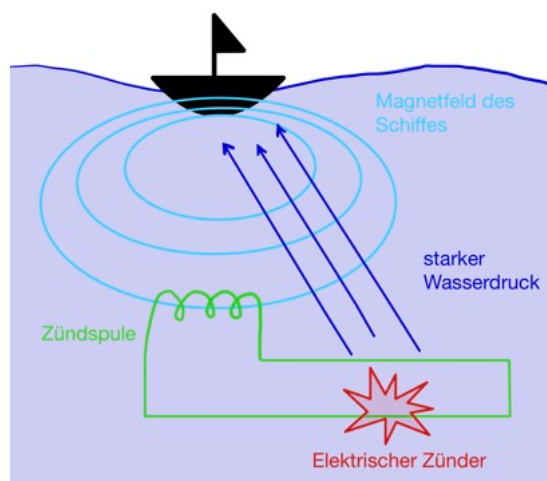


Abb. 15 - Magnetische Mine: Sarah Pappert

Magnetische Minen sind so aufgebaut, dass die Schiffe einen Magnetzünder überqueren, der einen etwas versetzten Torpedo aktiviert. Dieser trifft das Schiff nicht direkt, sondern übt über das fast nicht kombinierbare Wasser ganzflächig großen Druck auf das Schiff aus und verursacht damit gewaltigen Schaden, teilweise können die Schiffe dadurch sogar zerbrechen (siehe Abb. 15). Deswegen gelten Torpedos mit Magnetzünder als besonders gefährlich, aber sie können leicht umgangen werden, indem man die Schiffe entmagnetisiert. Einerseits findet direkt nach dem Bau

eine sogenannte „wicklungsfreie Entmagnetisierung“ statt, da keine stationären Wicklungen im Schiff angebracht werden. Es werden lediglich Drähte um das Schiff herum verlegt und ein Wechselfeld entlang der hysteresefreien Kurve erzeugt und die Magnetisierung so reduziert. Andererseits tragen viele Schiffe noch eine „Wicklungsentmagnetisierung“ an Board, die sich permanent im Rumpf befindet und in riskanten Gebieten aktiviert wird. Dabei wird durch Gleichstrom ein Feld erzeugt, was dem Feld des Schiffes im Betrag gleicht, aber im Vorzeichen entgegengesetzt ist. Unterhalb des Kiels wird somit ein nahezu feldfreier Zustand erzeugt, der das Schiff für magnetische Minen unsichtbar macht.

Zusätzlich bringt die Entmagnetisierung den Vorteil mit sich, dass der Schiffskompass nicht durch das Schiffsmagnetfeld und das Erdmagnetfeld durcheinander kommen kann.

Dieser kleine Exkurs soll verdeutlichen, dass Techniken zur Entmagnetisierung stark gefragt sind und deswegen auch Forschung in diesem Bereich stattfindet. Außerdem wollte ich aufzeigen, wie die Verfahren, die ich im kleinen getestet hab, im großen Stil Anwendung finden.

3.7 Modularisierung des Codes

Da wir mehrere Sensoren an denselben Raspberry Pi anschließen wollen und dabei den gleichen Bus nutzen, aber nicht mehrere Programme gleichzeitig auf den gleichen Bus zugreifen können, mussten wir uns etwas überlegen, wie wir alle Einzelprogramme zu einem großen Hauptprogramm zusammenfassen können, welches dann „allein“ auf den Bus zugreift und allen Sensoren ihre Befehle erteilt.

Dafür haben wir als erstes den Abstrich gemacht, dass wir nur die Rohwerte speichern und die Umrechnungen im Nachhinein durchführen, um Rechenleistung zu sparen. Wir haben nach Gemeinsamkeiten in unseren Codes gesucht, die wir in eine allgemein gültige Funktion umschreiben können und dann für jeden Sensor nur noch aufrufen müssen, wie zum Beispiel die Umwandlung in ein Zweierkomplement. Diese haben wir in der `function` Datei, quasi als Bibliothek, gespeichert. Spezielle Funktionen für die einzelnen Sensoren ebenfalls. In der `__init__` Datei werden die Sensoren angesprochen und die Funktionen aus unserer Bibliothek aufgerufen.

Der modularisierte Code ist wesentlich kürzer als mein ursprünglich geschriebener Code. Deswegen möchte ich kurz darauf eingehen, welche grundlegenden Befehle in dem modularisierten Code noch vorhanden sind.

Bei jedem Sensor wird zunächst ein Oversampling durchgeführt. Dabei wird das Signal mit einer höheren Abtastrate bearbeitet, als nötig. Dadurch werden, vereinfacht gesagt, Bereiche der Sensoren geprüft und aktiviert. Beim BME liegt die Besonderheit vor, dass er die Feuchtigkeit einzeln, aber Temperatur und Druck gemeinsam speichert. Außerdem können hierbei schon wichtige Werte zugewiesen werden, beim Magnetfeldsensor sind das Frequenz, Fieldrange und Modus. Im nächsten Schritt werden die Daten beider Sensoren als Einer-Komplement (nicht vorzeichenbehaftet) ausgelesen. Der BME verfügt über drei Register, weswegen für ihn die Funktion `fct.read_3byte()` Verwendung findet. Der HMC hat nur zwei Register, auf ihn wird `fct.read_word()` angewandt. Schließlich wird noch `write_file` genutzt, um die Messwerte zusammen mit Uhrzeit und Datum (aus `strftime`) in regelmäßigen Abständen in neue Dateien zu speichern, die jeweils auch die aktuelle Uhrzeit im Dateinamen tragen. Die Absicht ist auch hier, noch Daten retten zu können, falls ein Problem auftritt und deswegen Textdokumente unbrauchbar werden, wie bereits unter Punkt 3.3 angesprochen.

Zusammenfassend kann man sagen, dass die Kernelemente (Oversampling, Auslesen und Konvertieren der Daten, in Datei speichern) viel effizienter und kompakter gestaltet wurden und Umrechnungen komplett weg gestrichen wurden, wodurch der Code viel kürzer ist.

4 Missionsdurchführung

4.1 Aufbau der Sonde

Wir sind in der komfortablen Situation, dass bereits 2020 ein STRATO Ballon an unserer Uni gestartet wurde und wir somit einiges an Material, wie zum Beispiel die Styroporbox, in der sich das Payload befinden wird, übernehmen konnten.

In der Box werden sich drei Raspberry Pi's jeweils samt Powerbank befinden - einer für MuonPi, einer für die Steuerung und einer für alle Sensoren. Um alle Sensoren gleichzeitig an den I²C Bus anschließen zu können, wurden sie alle als Parallelschaltung verlötet. Hinzu kommt das Mikrobio-Experiment sowie Taschenwärmer, damit die Temperatur im Inneren nicht zu schnell und abrupt abfällt.

Der eigentliche Ballon sowie sonstiges Zubehör wurden auf <https://www.stratoflights.com> bestellt.



Abb. 16a - Aufbau der Sonde:
Sarah Pappert



Abb. 16b - Sonde von Außen:
Alex Stamm

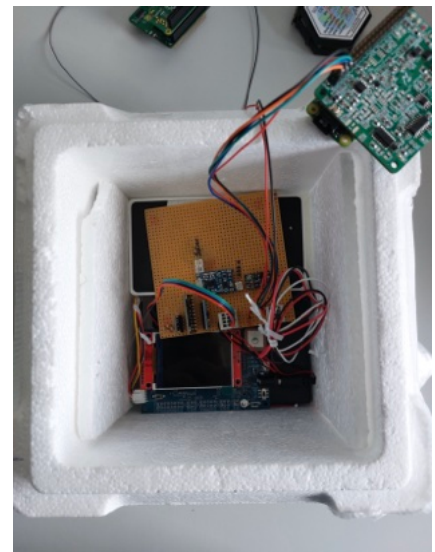


Abb. 16c - Sonde von Innen:
Alex Stamm

Meinen Sensor BME280 sieht man in violett in Abb. 16b und der HMC5889 ist der erste Sensor von links in Abb. 16c.

4.2 Start des Ballons

Der Start des Ballons verlief zunächst gut, allerdings haben wir kurz danach den Kontakt verloren. Am nächsten Tag wurde der Ballon von Funkern in Belgien über Gent gehört, im Anschluss daran gab es keine Sichtungen mehr und der Aufenthaltsort des Ballons ist nach wie vor unbekannt. Es wird demnächst einen weiteren Stratoballon-Start an unserer Uni geben, aber ich möchte schon mal ein paar Messergebnisse meines Sensors auf dem Erdboden vorstellen.

5 Auswertung des alternativen Experiments

5.1 Umwandlung des Magnetfeldsensors in einen Kompass

Unter Punkt 3.4 wurde bereits die Positionsabhängigkeit des Magnetfeldsensors besprochen, die für eine kontinuierliche Messung des „tatsächlichen“ Werts des Erdmagnetfelds eher hinderlich war. Nun kann man sich diese zu Nutze machen und ihn in seiner ursprünglichen Funktion als Kompass verwenden, indem man die Höhe über dem Erdboden konstant hält und den Sensor langsam um die eigene Achse dreht und in Ost-West sowie Nord-Süd-Ausrichtung kurz verharrt. Es ergibt sich ein Maximum im Norden und ein Minimum im Süden. Das macht Sinn, da mein Sensor nicht den Absolutbetrag des B-Feldes misst. Der induzierte Strom ist richtungsabhängig und, wie in Abschnitt 2.4 bereits angesprochen, hängt der Widerstand des AMR-Materials und dementsprechend unser Messwert vom Winkel zwischen Stromdichtevektor und Magnetfeldvektor des inneren Magnetfelds ab, somit liegt das Maximum nicht 90° zu Nord-Süd, sondern 180° , was genau den Messergebnissen entspricht.

Ich habe meinen Kompass mithilfe einer Powerbank mobil gestaltet und an mehreren Tagen zu meinem ständigen Begleiter gemacht und ihn mit in die JLU, THM und Innenstadt sowie in meine Heimat in Sachsen Anhalt genommen und möchte einerseits herausfinden, ob ich Richtungsänderungen anhand der Messwerte feststellen kann und andererseits, ob sich irgendwelche Änderungen in der Magnetfeldstärke (beispielsweise durch Ampeln oder durch die Experimente im physikalischen Institutsgebäude) ergeben.

5.2 Auswertung der Messergebnisse

5.2.1 Weg zur JLU

Am Montag, den 20.06.2022, bin ich um 10.10 Uhr mit dem Fahrrad in Richtung Süd-Ost zum Naturwissenschaftscampus der Justus-Liebig-Universität gestartet. In den Messwerten (Abb. 18a) erkennt man, dass zu diesem Zeitpunkt $B \approx 700mG$ beträgt. Ab 10.11 Uhr springt B auf $\approx 1700mG$, wo ich in Richtung Nord-West gefahren bin, und fällt 10.12 Uhr wieder auf $\approx 1300mG$, als ich in Richtung Nord-Ost gefahren bin. 10.13 Uhr sinkt B kurzzeitig auf $\approx 700mG$, weil ich in südliche Richtung gefahren bin, aber kurz danach steigt es direkt wieder auf $\approx 1300mG$, als ich wieder in nord-östliche Richtung gefahren bin. Dieser Wert bleibt relativ konstant, da ich meine Richtung nun beibehalte. Erst 10.18 Uhr fahre ich wieder kurzzeitig nach Süden, wodurch wieder $B \approx 700mG$ beträgt. Danach befindet sich der Wert wieder bei $\approx 1400mG$, da ich in Richtung Nord-Ost unterwegs bin. Bis ich 10.24 schließlich an der Uni angekommen bin und mein Fahrrad in westlicher Richtung anschließe, was sich in $B \approx 800mG$ widerspiegelt.

Bis 10.46 Uhr bleibt der Wert des B-Feldes weitestgehend konstant, da ich in dieser Zeit eine Vorlesung hatte und der Sensor stillstand. Zwischen 10.46 und 12.11 Uhr bin ich nochmal nach Hause und wieder zurück in die Uni gefahren, was wir an deutlichen Ausschlägen im Graphen erkennen können. Von 13.46 bis 14 Uhr bin ich durch die unterschiedlichen Flure im Institutsgebäude gegangen, wo ich keine Erhöhung des B-Feldes feststellen konnte. Danach schließen sich ein Nachhauseweg, eine 90-minütige Freistunde, ein Weg zur Uni (dieses Mal mit dem Bus), eine 90-minütige Vorlesung und schließlich 17.55 Uhr wieder ein Nachhauseweg (nun zu Fuß) an. Da sich für gleiche Richtungen gleiche Werte ergeben, kann man auch nur aus den Messwerten auf die zurückgelegte Strecke schließen.

Diese Messung belegt sehr gut, dass im Norden ein Maximum und im Süden ein Minimum gemessen wird. Es gibt einzelne Ausreißer, die bis zu $B \approx 8000mG$ betragen. Grund dafür muss das Aufsetzen meines Rucksacks und dadurch eine Erschütterung am Sensor sein, es lässt sich anhand der Uhrzeiten keine andere Erklärung dafür finden. Bei den Messungen in der THM trat dieser Effekt ebenfalls auf, aber danach konnte ich ihn nicht nochmal reproduzieren. Diese Ausreißer sowie gelegentliche Ungenauigkeiten („Zacken“) lassen sich auch auf die in Punkt 3.5 besprochene Magnetisierung des Sensors zurückführen.

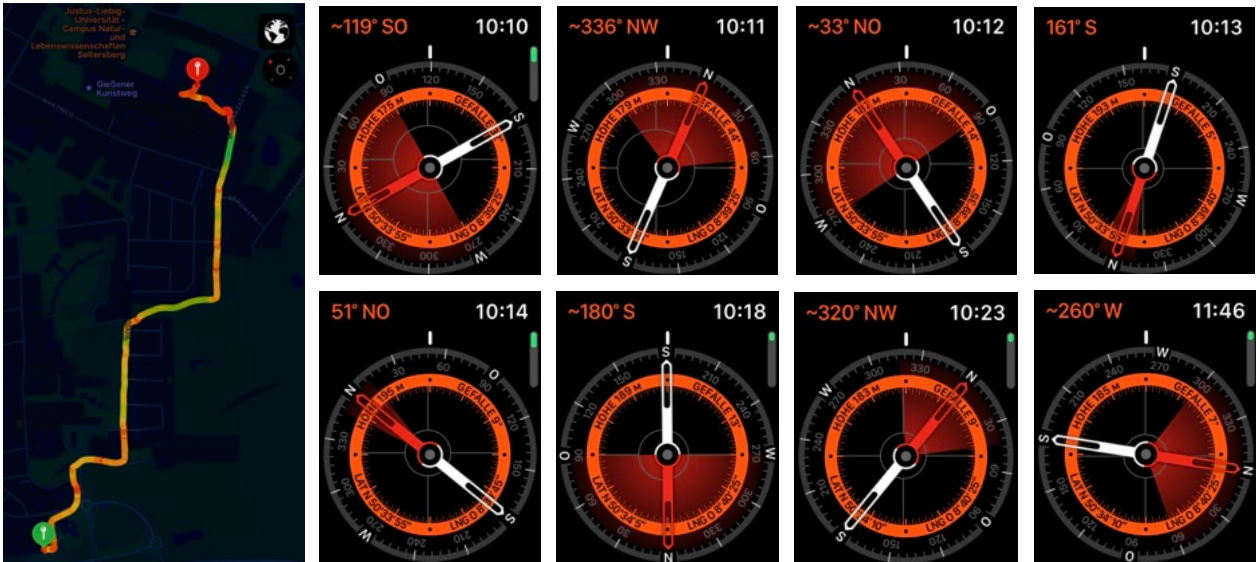


Abb. 17 - Streckenverlauf zur JLU und mit zweitem Kompass gemessene Richtungen: Sarah Pappert via AppleWatch Series 5

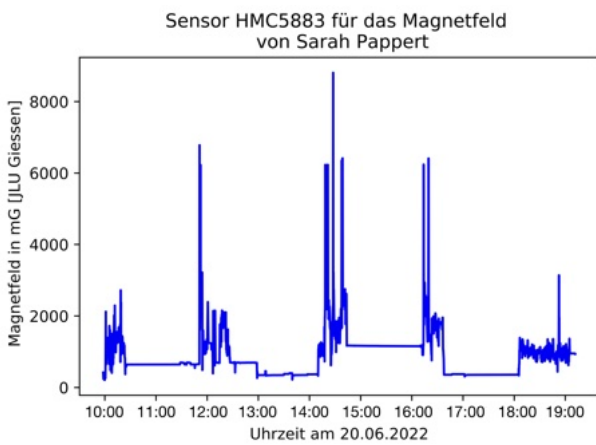


Abb. 18 - gesamter Verlauf der magnetischen Flussdichte B

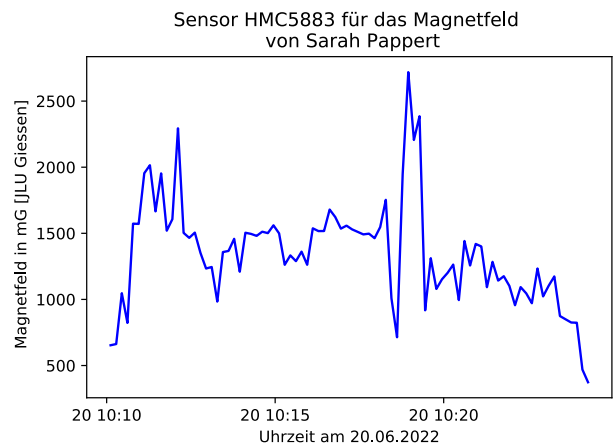


Abb. 18a - Weg zur Uni mit dem Fahrrad

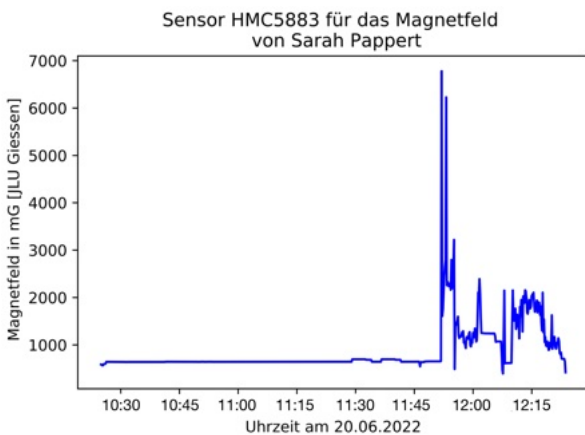


Abb. 18b - Vorlesung und Hin- und Herfahrt zur Uni mit dem Fahrrad

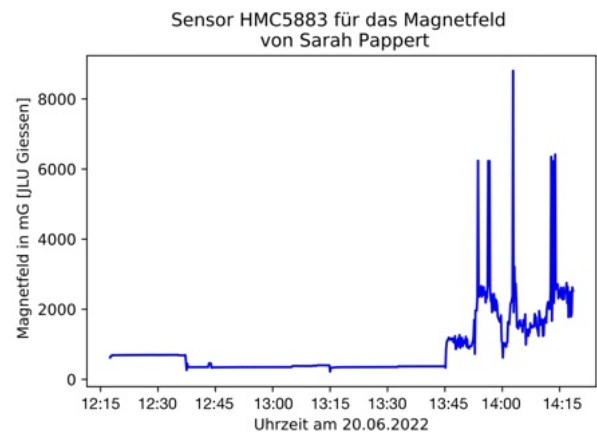


Abb. 18c - Vorlesung, Weg durch die Uni und Nachhauseweg mit Fahrrad

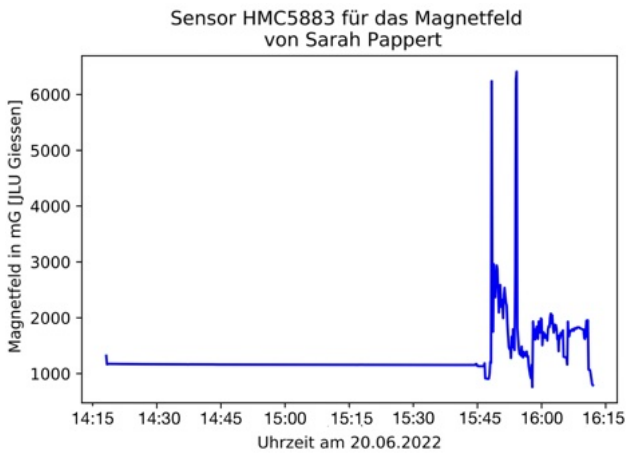


Abb. 18d - zu Hause und Weg zur Uni mit dem Bus

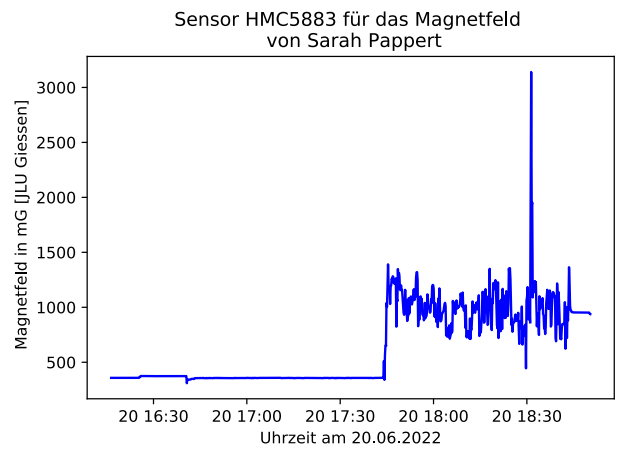


Abb. 18e - Vorlesung und Nachhauseweg zu Fuß

5.2.2 Weg zur THM und in die Innenstadt

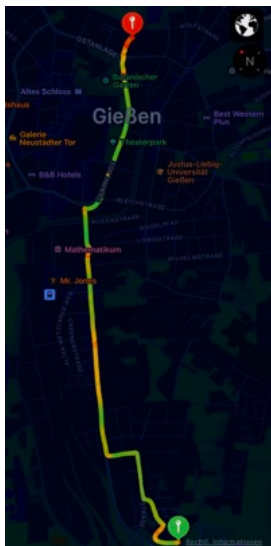


Abb. 19 - Streckenverlauf zur THM: Sarah Pappert via AppleWatch Series 5

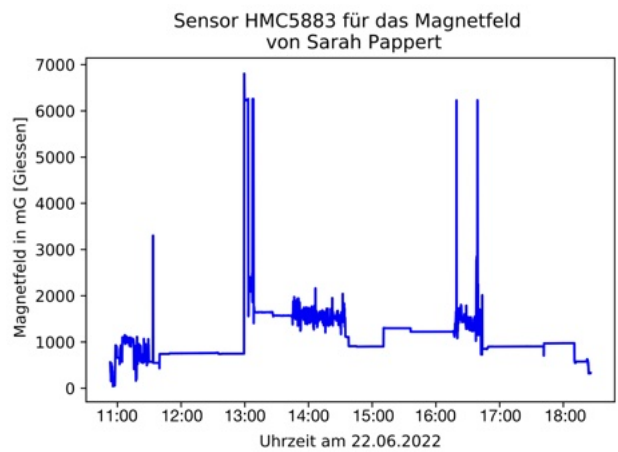


Abb. 20 - gesamter Verlauf der magnetischen Flussdichte B

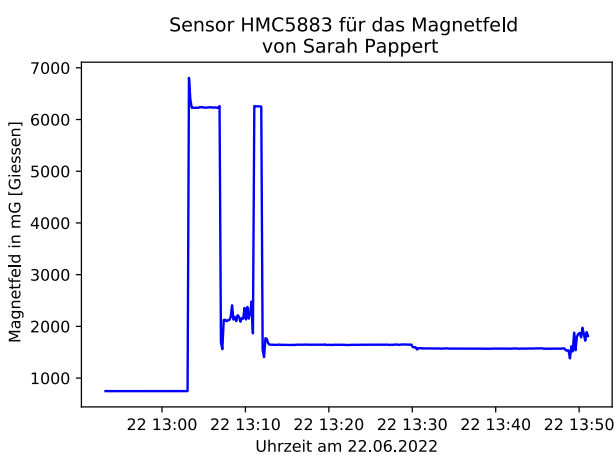


Abb. 20a - THM Mensa

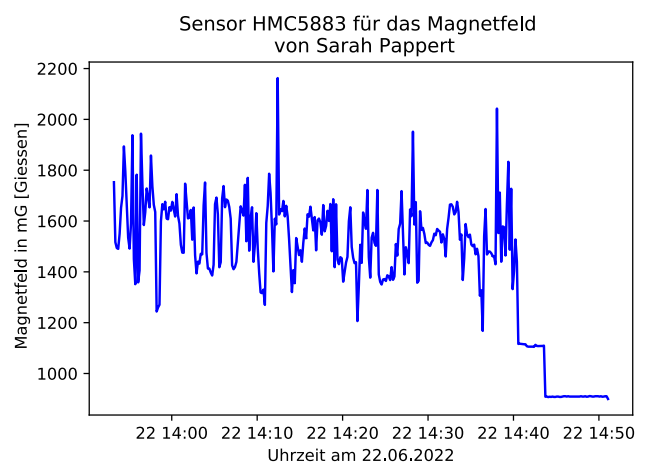


Abb. 20b - Weg durch die Gießener Innenstadt

Wie man in Abb. 20 sieht, bin ich mit dem Fahrrad von 11.04 - 11.17 Uhr zur THM gefahren und hatte von 11.30 - 13 Uhr eine Vorlesung. In dieser Zeit ist das Magnetfeld konstant. Im Anschluss an die Vorlesung bin ich in der Mensa essen gegangen, hier sieht man wieder starke Ausreißer bei $\approx 6500mG$, von denen im Gegensatz zu Abb. 18 gleich mehrere auftreten. Auffällig ist, dass

diese Ausreißer immer am Ende einer Vorlesung auftreten, was dafür spricht, dass es etwas mit dem Aufsetzen des Rucksacks zu tun hat. Die Kabel sind lang und locker, außerdem war mein iPad in der Tasche, es ist durchaus möglich, dass deswegen etwas durch Bewegung induziert wurde. Allerdings sind diese Ausreißer über einen Zeitraum von zehn Minuten verteilt. Es ist fraglich, ob Geräte in der Mensa so ein starkes Magnetfeld erzeugen, vermutlich gibt es eine andere Ursache. Von 13.56 - 14.38 Uhr war ich in der Stadt unterwegs, was man anhand der Schwankungen der magnetischen Flussdichte aufgrund der Richtungsänderungen in Abb. 20b erkennen kann. Von 16.32 - 16.49 Uhr bin ich schließlich mit dem Fahrrad wieder nach Hause gefahren.

5.2.3 Feldweg in Lutherstadt Wittenberg

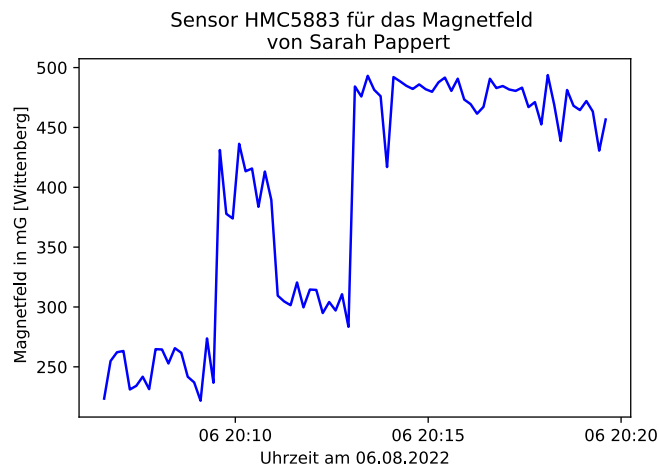


Abb. 21 - Richtungsänderungen auf dem Feldweg: Sarah Pappert

Um die Richtungsabhängigkeit des Sensors nochmal zu verdeutlichen, habe ich mir einen Weg, der genau entlang einer Nord-Süd-Achse ausgerichtet ist, gesucht. Zunächst bin ich von 20.06 bis 20.09 Uhr in südliche Richtung gelaufen, dann nach Westen (leicht nord-westlich), ab 20.11 Uhr nach Osten (leicht süd-östlich) und ab 20.13 Uhr schließlich nach Norden. Es sind zwar leichte Schwankungen in den Messwerten, aber dennoch erkennt man sehr deutlich die Stufenform der Kurve - jede Richtung hat eine andere Magnetfeldstärke. Hier ist auch nochmal verdeutlicht, dass der Wert im Norden am höchsten ist, im Westen etwas geringer, im Osten nochmal geringer und im Süden schließlich minimal. Wäre ich genau im 90°-Winkel abgelenkt, wären Osten und Westen fast identisch.

5.2.4 Ampel

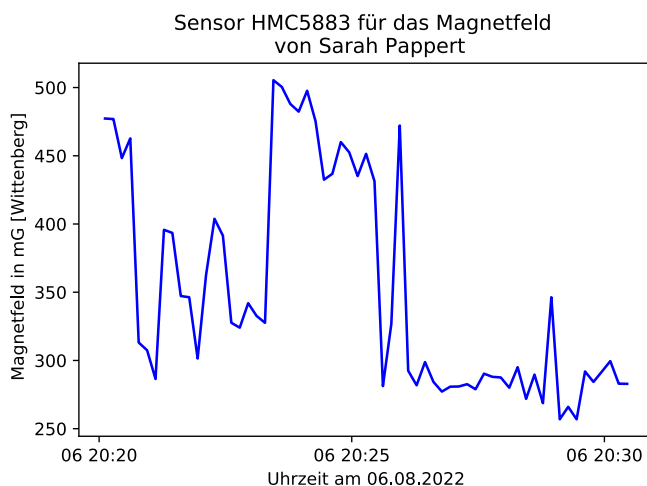


Abb. 22 - Ampel: Sarah Pappert



Abb. 23 - Ampelkreuzung, Position beim blauen Kreis: Apple Karten

Für dieses Experiment habe ich mich auf eine Verkehrsinsel einer großen Ampelkreuzung direkt neben die Induktionsschleife gestellt. An den Ausschlägen in der magnetischen Flussdichte erkennt man, wann Autos an mir vorbeigefahren sind. Um 20.23 Uhr hat sich beispielsweise eine Schlange aus 5 Autos an der Ampel gebildet, was sich in einem hohen und breiten Peak widerspiegelt. Die vier weiteren größeren Ausschläge wurden von einzelnen Autos verursacht, bei $B \approx 290mG$ liegt bei dieser Messung das Grundniveau ohne Autos.

Autos bestehen aus großen Mengen ferromagnetischer Stoffe, was im Funktionsprinzip einer Ampel genutzt wird: bewegt sich der Ferromagnet in das Magnetfeld der Leiterschleife der Ampel, wird ein Strom induziert und die Ampel schaltet um. Als solcher Ferromagnet ist ein Auto selbst auch von einem Magnetfeld umgeben, was ich in diesem Versuch messen konnte. Es beträgt ungefähr $\Delta B < 100mG$, was $\frac{1}{5}$ des natürlichen Erdmagnetfelds entsprechen würde, in der Realität ist der Wert aber nochmal deutlich geringer, weil diese Messung von vielen Faktoren (Leiterschleife, Geschwindigkeit der Autos, eventuell leichte Bewegung / Drehung des Sensors) beeinflusst wird.

5.2.5 Bahngleise

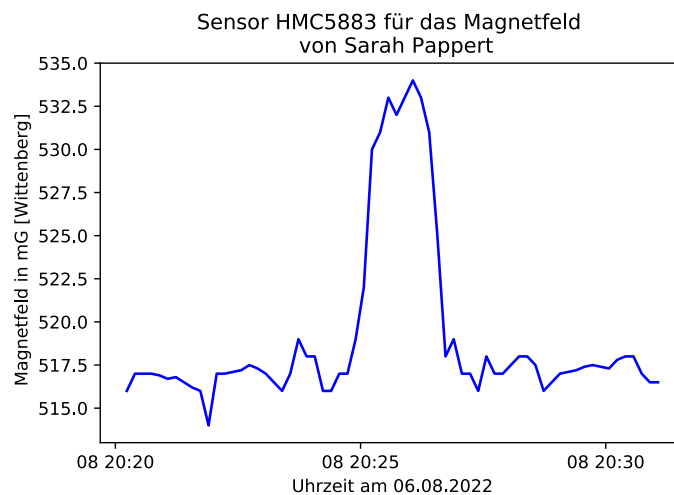


Abb. 24 - Magnetfeld eines vorbeifahrenden Zugs

Für diesen Versuch habe ich den Sensor in der Nähe von Bahngleisen positioniert. Zunächst ergibt sich die typische Stärke des Erdmagnetfelds, aber wenn ein Zug vorbeifährt, was zwischen 20.25 - 20.27 Uhr der Fall ist, misst man ein stärkeres B-Feld. In diesem Versuch beträgt die Erhöhung $\Delta B \approx 15mG$, was sehr gering ist. Zudem habe ich in der Nähe eines Bahnhofs gemessen, wo der Zug wahrscheinlich seine Wirbelstrombremsen betätigt hat, um langsamer zu werden. Ansonsten ist der Zug, wie auch das Auto, ein Ferromagnet und somit von einem Magnetfeld umgeben, außerdem nutzt er die Oberleitung. Deswegen entspricht ein höheres B bei der Vorbeifahrt den Erwartungen.

5.2.6 Telefonleitung

Wie in Tab. 3 bereits gezeigt, sind Telefon- und Hochspannungsleitungen von einem sehr starken Magnetfeld umgeben, was unmittelbar an der Leitung etwa dieselbe Stärke wie das natürliche Erdmagnetfeld hat. In diesem Versuch möchte ich herausfinden, wie schnell das Magnetfeld mit zunehmendem Abstand von der Leitung abnimmt und ob man als normaler Passant einem signifikant höheren Magnetfeld in der Nähe von Oberleitungen ausgesetzt ist.

Da zwischen B und r eine indirekte Proportionalität vorliegt, müsste B theoretisch in Form einer Hyperbel abnehmen.

Um den Versuch durchzuführen, habe ich 15m von der Leitung entfernt mit meiner Messung begonnen und mich der Leitung langsam genährt, bis ich direkt unter der Leitung war. Dort habe ich den Sensor in verschiedene Höhen gehalten und mich dann wieder von der Leitung entfernt. Deswegen sieht man in Abb. 25, dass der Peak der B-Feld-Kurve etwas in die Breite gezogen ist. In Abb. 25 a und b ist dargestellt, zu welchen Zeiten der horizontale und zu welchen Zeiten der vertikale Abstand verringert wurde, der andere ist jeweils konstant. Dadurch ergibt sich für das B-Feld eine nahezu glockenförmige Kurve.

Festzustellen ist, dass B in unserem Messbereich nur um $\approx 50mG$ ansteigt, was $\frac{1}{10}$ des natürlichen Erdmagnetfelds entspricht. Dies zeigt, dass die Flussdichte sehr schnell abnimmt und man in normalem Abstand zur Leitung (mindestens 3m vertikal und 1m horizontal) einem Magnetfeld ausgesetzt ist, was nur sehr geringfügig größer als das natürlich vorhandene ist und somit überhaupt keine Gefahr darstellt.

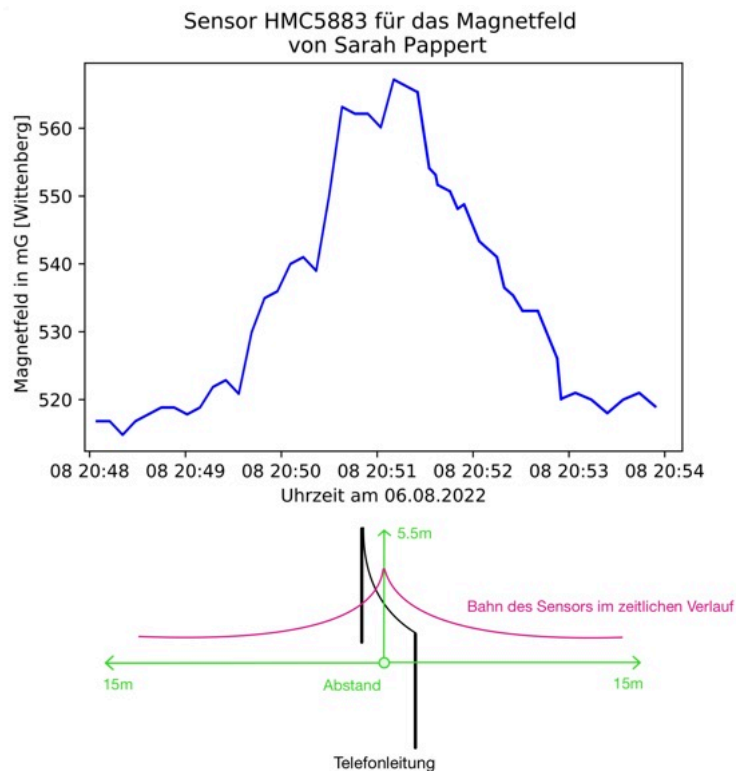


Abb. 25 - B-Feld der Telefonleitung im Vergleich zu resultierendem Abstand des Sensors: Sarah Pappert

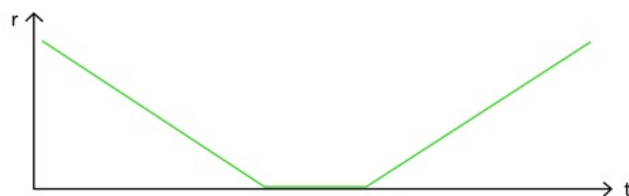


Abb. 25a - horizontaler Abstand in Abhängigkeit von der Zeit

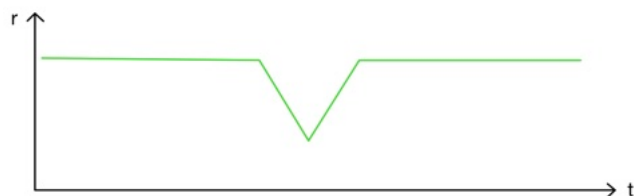


Abb. 25b - vertikaler Abstand in Abhängigkeit von der Zeit

6 Fazit

Abschließend kann ich sagen, dass ich an einem spannenden Projekt mit faszinierenden Möglichkeiten und wertvollen Erfahrungen teilgenommen habe. Unsere STRATO Mission verlief nicht wie geplant und einige Hypothesen bleiben unbewiesen, aber der Wissenszuwachs ist dennoch enorm. Ich habe durch dieses Projekt programmieren gelernt und viel über Magnetfelder erfahren sowie praktische Möglichkeiten zur Magnetisierung und Entmagnetisierung kennengelernt.

Außerdem war einiges an Kreativität gefragt, um zumindest einige Aspekte der Theorie praktisch nachzuweisen und ich bin mit diesen Ergebnissen sehr zufrieden, sie haben die Erwartungen erfüllt und lassen sich hervorragend erklären. Ich konnte letztendlich sehr genau den Wert des Erdmagnetfelds bestimmen, da die auf den Sensor angewandten Strategien wie gewünscht funktioniert haben, und auch die Messungen an den künstlichen Magnetfeldern sowie die Umwandlung des Sensors in einen Kompass verliefen genau wie geplant.

Nicht zu vergessen ist auch, dass wir gelernt haben, als Team miteinander zu arbeiten und uns dementsprechend zu organisieren.

Als zentrale Ergebnisse dieser Arbeit sind die Programmierung der Sensoren BME280 sowie HMC5883, deren Testung und Optimierung bis hin zu plausiblen Ergebnissen, die Recherche über Magnetfelder und deren technische Anwendungen, das Entmagnetisieren von Bauteilen, die Vermessung der Magnetfelder von Ampeln, Autos, Zügen und Telefonleitungen, der Bau eines Kompasses sowie die Planung und Organisation des Ballonstarts zu nennen.

Eine Schwierigkeit, die sich bei der Durchführung des Ballonflugs ergeben hätte, wäre die Positionsabhängigkeit meines Magnetfeldsensors gewesen, da sich beim Start bereits abgezeichnet hat, dass das Payload sich um die eigene Achse gedreht hat. Die Konsequenzen wurden bereits in Abschnitt 3.5 diskutiert. Für folgende Missionen würde ich deshalb empfehlen, keinen AMR-Sensor zu verwenden, sondern lieber auf eine Hall-Sonde zurückzugreifen.

Zwischenzeitlich stand außerdem zur Debatte, einen Flugzeug-Transponder mit an Board zu nehmen, um den Ballon selbst in großen Höhen live verfolgen zu können und ihn auch für Flugzeuge auf dem Radar sichtbar zu machen. Ich persönlich würde diesen Plan beim nächsten Start in die Tat umsetzen, um auf der ganz sicheren Seite zu sein, da der Transponder unabhängig von unseren anderen Messgeräten funktioniert.

Alles in allem würde ich unsere Mission somit als Erfolg bezeichnen. Wir haben vermutlich mit der spektakulärsten Flugbahn eines Wetterballons aller Zeiten Geschichte geschrieben und einiges gelernt, wodurch die nächste Mission mit Sicherheit nach Plan verlaufen wird.

Ich für meinen Teil konnte durch Improvisieren trotzdem einige Messungen an Magnetfeldern durchführen und habe tolle Ergebnisse erhalten und sowohl die natürlichen als auch künstlichen Felder auf dem Erdboden näher kennengelernt. In Bezug auf meine Einleitung kann ich nun bestätigen, dass die Messung eines höhenabhängigen Verlaufs des Erdmagnetfelds tatsächlich einige Schwierigkeiten mit sich bringt, die aber durchaus zu meistern sind.

Ich bin gespannt auf die Ergebnisse der zweiten Mission und auf die zukünftige Forschung am Erdmagnetfeld. Wer weiß, welche Überraschungen und neuen Möglichkeiten dieses noch für uns birgt...



Sarah Pappert

Lutherstadt Wittenberg, den 09.08.2022


```

# Read blocks of calibration data from EEPROM
# See Page 22 data sheet
cal1 = bus.read_i2c_block_data(addr, 0x88, 24)
cal2 = bus.read_i2c_block_data(addr, 0xA1, 1)
cal3 = bus.read_i2c_block_data(addr, 0xE1, 7)

# Convert byte data to word values
dig_T1 = getUShort(cal1, 0)
dig_T2 = getShort(cal1, 2)
dig_T3 = getShort(cal1, 4)

dig_P1 = getUShort(cal1, 6)
dig_P2 = getShort(cal1, 8)
dig_P3 = getShort(cal1, 10)
dig_P4 = getShort(cal1, 12)
dig_P5 = getShort(cal1, 14)
dig_P6 = getShort(cal1, 16)
dig_P7 = getShort(cal1, 18)
dig_P8 = getShort(cal1, 20)
dig_P9 = getShort(cal1, 22)

dig_H1 = getUChar(cal2, 0)
dig_H2 = getShort(cal3, 0)
dig_H3 = getUChar(cal3, 2)

dig_H4 = getChar(cal3, 3)
dig_H4 = (dig_H4 << 24) >> 20
dig_H4 = dig_H4 | (getChar(cal3, 4) & 0x0F)

dig_H5 = getChar(cal3, 5)
dig_H5 = (dig_H5 << 24) >> 20
dig_H5 = dig_H5 | (getUChar(cal3, 4) >> 4 & 0x0F)

dig_H6 = getChar(cal3, 6)

# Wait in ms (Datasheet Appendix B: Measurement time and current calculation)
wait_time = 1.25 + (2.3 * OVERSAMPLE_TEMP) + ((2.3 * OVERSAMPLE_PRES) + 0.575) + ((2.3 * OVERSAMPLE_HUM)+0.575)
time.sleep(wait_time/1000) # Wait the required time

# Read temperature/pressure/humidity
data = bus.read_i2c_block_data(addr, REG_DATA, 8)
pres_raw = (data[0] << 12) | (data[1] << 4) | (data[2] >> 4)
temp_raw = (data[3] << 12) | (data[4] << 4) | (data[5] >> 4)
hum_raw = (data[6] << 8) | data[7]

#Refine temperature
var1 = (((temp_raw>>3)-(dig_T1<<1))*(dig_T2)) >> 11
var2 = (((temp_raw>>4) - (dig_T1)) * ((temp_raw>>4) - (dig_T1))) >> 12) * (dig_T3)) >> 14
t_fine = var1+var2
temperature = float(((t_fine * 5) + 128) >> 8);

# Refine pressure and adjust for temperature
var1 = t_fine / 2.0 - 64000.0
var2 = var1 * var1 * dig_P6 / 32768.0
var2 = var2 + var1 * dig_P5 * 2.0
var2 = var2 / 4.0 + dig_P4 * 65536.0
var1 = (dig_P3 * var1 * var1 / 524288.0 + dig_P2 * var1) / 524288.0
var1 = (1.0 + var1 / 32768.0) * dig_P1
if var1 == 0:
    pressure=0
else:
    pressure = 1048576.0 - pres_raw
    pressure = ((pressure - var2 / 4096.0) * 6250.0) / var1
    var1 = dig_P9 * pressure * pressure / 2147483648.0
    var2 = pressure * dig_P8 / 32768.0
    pressure = pressure + (var1 + var2 + dig_P7) / 16.0

# Refine humidity
humidity = t_fine - 76800.0
humidity = (hum_raw - (dig_H4 * 64.0 + dig_H5 / 16384.0 * humidity)) * (dig_H2 / 65536.0 * (1.0 + dig_H6 /
67108864.0 * humidity * (1.0 + dig_H3 / 67108864.0 * humidity)))
humidity = humidity * (1.0 - dig_H1 * humidity / 524288.0)
if humidity > 100:
    humidity = 100
elif humidity < 0:
    humidity = 0

return temperature/100.0,pressure/100.0,humidity

def append_new_line(file_name, text_to_append):
    """Append given text as a new line at the end of file"""
    # Open the file in append & read mode ('a+')
    with open(file_name, "a+") as file_object:
        # Move read cursor to the start of file.
        file_object.seek(0)
        # If file is not empty then append '\n'
        data = file_object.read(100)
        if len(data) > 0:
            file_object.write("\n")
        # Append text at the end of file
        file_object.write(text_to_append)

```



```

import argparse

# Global data
# I2C bus (1 at newer Raspberry Pi, older models use 0)
bus = smbus.SMBus(1)
# I2C address of HMC5883
address = 0x1e

# Read block data from HMC5883
def read_data():
    return bus.read_i2c_block_data(address, 0x00)

# Convert val to signed value
def twos_complement(val, len):
    if (val & (1 << len - 1)):
        val = val - (1<<len)
    return val

# Convert two bytes from data starting at offset to signed word
def convert_sw(data, offset):
    return twos_complement(data[offset] << 8 | data[offset+1], 16)

# Write one byte to HMC5883
def write_byte(adr, value):
    bus.write_byte_data(address, adr, value)

def append_new_line(file_name, text_to_append):
    """Append given text as a new line at the end of file"""
    # Open the file in append & read mode ('a+')
    with open(file_name, "a+") as file_object:
        # Move read cursor to the start of file.
        file_object.seek(0)
        # If file is not empty then append '\n'
        data = file_object.read(100)
        if len(data) > 0:
            file_object.write("\n")
        # Append text at the end of file
        file_object.write(text_to_append)

def applygain(value, gain):
    float (val)*float(gain)
    return val

# Main
def main():
    # Init HMC5883
    write_byte(0, 0b01100000) # Rate: 8 samples @ 15Hz
    write_byte(1, 0b01100000) # Sensor Field range 2.5 Ga -> 1,52mG/LSB
    write_byte(2, 0b00000000) # Mode: Continuous sampling

    while(1==1):
        t = time.strftime("%d.%m.%Y %H:%M:%S")
        myname = "hmc_" + str(t) + ".txt"
        print myname

        for i in range(0,30):
            # read data from HMC5883
            data = read_data()

            # get x,y,z values of magnetic induction
            bx = convert_sw(data, 3) # x
            by = convert_sw(data, 7) # y
            bz = convert_sw(data, 5) # z
            b = math.sqrt(float(bx*bx) + float(by*by) + float(bz*bz))

            #apply gain factor from datasheet according #to Sensor field range
            xneu=bx*1.52
            zneu=bz*1.52
            yneu=by*1.52
            neu=math.sqrt(xneu*xneu + yneu*yneu + zneu*zneu)

            print "bxneu : ", xneu, "mG", " - ", xneu*100.0, "nT"
            print "byneu : ", yneu, "mG", " - ", yneu*100.0, "nT"
            print "bzneu : ", zneu, "mG", " - ", zneu*100.0, "nT"
            print "bneu : ", neu, "mG", " - ", neu*100.0, "nT"

            datetime = time.strftime("%d.%m.%Y,%H:%M:%S")

            data_as_string = str(datetime) + "," + str(xneu) + "," + str(yneu) + "," + str(zneu) + "," + str(neu)
            append_new_line(myname, data_as_string)
            append_new_line("hmc_measurements.txt", data_as_string)

            time.sleep(10)

        time.sleep(10)

if __name__ == '__main__':
    main()

```

7.4 Python Code zur Datenauswertung

```
import matplotlib.pyplot as plt
import csv
import numpy as np
import datetime

#Daten einlesen und vorverarbeiten
dataset = np.genfromtxt("measurements_hmc5883_positionen.txt", delimiter = ",")
customdate = datetime.datetime(2022, 4, 21, 19, 46, 27) #Zeitstempel des ersten Messwerts: Jahr, Monat, Tag,
Stunde, Minute, Sekunde
dates = [customdate + datetime.timedelta(seconds=10*i) for i in range(len(dataset[:,1]))] #10 entspricht dem
Abstand zwischen den Messungen in Sekunden

#plotten
plt.plot(dates, dataset[:,4], 'g')
plt.title("Sensor HMC5883 für das Magnetfeld \n von Sarah Pappert", ha = 'center')
plt.xlabel("Uhrzeit \n in Stunde : Minute : Sekunde \n am 21.04.2022")
plt.ylabel(" Magnetfeld in mG \n [Positionen]")

#Peaks markieren
a = datetime.datetime(2022, 4, 21, 19, 50, 17)
plt.scatter([a,],[1015,], 50, color='blue')
plt.annotate(r'Maximum', xy=(a, 1015), xycoords='data', xytext=(+5, -1), textcoords='offset points', fontsize=8)

b = datetime.datetime(2022, 4, 21, 19, 54, 47)
plt.scatter([b,],[92.8,], 50, color='blue')
plt.annotate(r'Minimum', xy=(b, 92.8), xycoords='data', xytext=(+6, +0), textcoords='offset points', fontsize=8)

c = datetime.datetime(2022, 4, 21, 19, 58, 27)
plt.scatter([c,],[498.4,], 50, color='blue')
plt.annotate(r'Optimum', xy=(c, 498.4), xycoords='data', xytext=(-19, +7), textcoords='offset points', fontsize=8)

d = datetime.datetime(2022, 4, 21, 20, 2, 37)
plt.scatter([d,],[1007.8,], 50, color='blue')
plt.annotate(r'Maximum', xy=(d, 1007.8), xycoords='data', xytext=(-45, +0), textcoords='offset points',
fontsize=8)

#PDF erstellen
plt.savefig("Magnet_Test.pdf")
```

7.5 Modularisierter Code

© Nico Krug

__init__.py :

```
import time
from datetime import datetime
import smbus
from ADS1x15 import ADS1115
import fct

def init():

    #----- mpu6050 -----

    # Aktivieren, um das Modul ansprechen zu koennen
    fct.write_byte(smbus.SMBus(1), 0x68, 0x6b, 0x00)
    # Empfindlichkeit Gyro setzen
    fct.write_byte(smbus.SMBus(1), 0x68, 0x1b, 0b00001000)
    # Empfindlichkeit Acce setzen
    fct.write_byte(smbus.SMBus(1), 0x68, 0x1c, 0b00000000)

    #----- dfrobot-sen0321 -----

    fct.write_byte(smbus.SMBus(1), 0x72, 0x03, 0x00) # automatic mode

    #----- bme280 -----

    fct.write_byte(smbus.SMBus(1), 0x76, 0xF2, 0b00000010) #oversampling humidity
    fct.write_byte(smbus.SMBus(1), 0x76, 0xF4, 0b01001011) #oversampling temperature,
oversampling pressure, mode
```

```

#----- hmc5883 -----

fct.write_byte(smbus.SMBus(1), 0x1e, 0x0, 0b01110000) # Rate: 8 samples @ 15Hz
fct.write_byte(smbus.SMBus(1), 0x1e, 0x1, 0b01100000) # Sensor field range: 8.1 Ga
fct.write_byte(smbus.SMBus(1), 0x1e, 0x2, 0b00000000) # Mode: Continuous sampling

#####

init()

while 1:

#----- mpu6050 -----

    gyro = [fct.read_word_2c(smbus.SMBus(1), 0x68, 0x3d), fct.read_word_2c(smbus.SMBus(1), 0x68, 0x3b),
fct.read_word_2c(smbus.SMBus(1), 0x68, 0x3f)]
    acce = [fct.read_word_2c(smbus.SMBus(1), 0x68, 0x43), fct.read_word_2c(smbus.SMBus(1), 0x68, 0x45),
fct.read_word_2c(smbus.SMBus(1), 0x68, 0x47)]
    gyro_grad = gyro[1]/16384.0*180

    fct.write_file(fct.filename("gyro_raw", "csv", 5), "raw", gyro, 3)
    fct.write_file(fct.filename("acce_raw", "csv", 5), "raw", acce, 3)

#----- guva-s12sd -----

    snt = 2
    uvse = fct.adc_read(ADS1115(0x4a), 0, snt)

    fct.write_file(fct.filename("uvse_raw", "csv", 5), "raw", uvse)

#----- dfrobot-sen0321 -----

    ozon = fct.read_word(smbus.SMBus(1), 0x72, 0x09)

    fct.write_file(fct.filename("ozon_raw", "csv", 5), "raw", ozon)

#----- bme280 -----

    pres = fct.read_3byte(smbus.SMBus(1), 0x76, 0xF7)
    temp = fct.read_3byte(smbus.SMBus(1), 0x76, 0xFA)
    humi = fct.read_word(smbus.SMBus(1), 0x76, 0xFD)

    fct.write_file(fct.filename("pres_raw", "csv", 5), "raw", pres)
    fct.write_file(fct.filename("temp_raw", "csv", 5), "raw", temp)
    fct.write_file(fct.filename("humi_raw", "csv", 5), "raw", humi)

#----- hmc5883 -----

    magn = [fct.read_word(smbus.SMBus(1), 0x1E, 0x03), fct.read_word(smbus.SMBus(1), 0x1E, 0x07),
fct.read_word(smbus.SMBus(1), 0x1E, 0x05)]

    fct.write_file(fct.filename("magn_raw", "csv", 5), "raw", magn, 3)

    time.sleep(.1)

```

function.py :

```

import time
from datetime import datetime
import smbus
from ADS1x15 import ADS1115

```

```

#----- i2c Befehle -----

def read_byte(bus, adr, reg):
    reg=register sensor
    return bus.read_byte_data(adr, reg)
#bus=i2c bus nummer; adr=adresse sensor;
#bus=aus smbus SMBus(1)

def read_byte_2c(bus, adr, reg):
    tmp = read_byte(bus, adr, reg)
    if (tmp >= 0x80):
        return -((255 - tmp) + 1)
    else:
        return tmp
#als 2er-komplement auslesen

def read_word(bus, adr, reg):
    h = bus.read_byte_data(adr, reg)
    l = bus.read_byte_data(adr, reg+1)
    tmp = (h << 8) + l
    return tmp
#als 1er-komplement auslesen

def read_word_2c(bus, adr, reg):
    tmp = read_word(bus, adr, reg)
    if (tmp >= 0x8000):
        return -((65535 - tmp) + 1)
    else:
        return tmp
#als 2er-komplement auslesen

def read_3byte(bus, adr, reg):
    h = bus.read_byte_data(adr, reg)
    l = bus.read_byte_data(adr, reg+1)
    x = bus.read_byte_data(adr, reg+2)
    tmp = (h << 16) + (l << 8) + x
    return tmp

def read_3byte_2c(bus, adr, reg):
    tmp = read_3byte(bus, adr, reg)
    if (tmp >= 0x800000):
        return -((16777215 - tmp) + 1)
    else:
        return tmp

def write_byte(bus, adr, reg, cmd):
    return bus.write_byte_data(adr, reg, cmd)
#cmd=command

#----- adc Befehle -----

def adc_read(adc, num, snt):
    return adc.read_adc(num, snt, 128)
#snt=sensitivity

#----- Datei Befehle -----

def write_file(nam, dir, val, num = 1):
    pat = "/home/pi/strato/" + dir + "/" + nam
    datei = open(pat, 'a')
    datei.write("\n")
    datei.write(datetime.utcnow().strftime('%Y:%m:%d; %H:%M:%S.%f')[:-3])
    if (num == 1):
        datei.write(";%15d" % (val))
    else:
        i = 0
        while (i < num):
            datei.write(";%15d" % (val[i]))
            i += 1
    datei.write(";")
    datei.close()
    return 1

def filename(nam, typ, tim):
    min = int(datetime.utcnow().strftime('%M')) + int(datetime.utcnow().strftime('%H')) * 60
    while (min % tim != 0):
        min -= 1
    hou = int(min / 60)
    if(int(hou / 10) == 0):
        houstr = '0' + str(hou)
    else:
        houstr = str(hou)
    min = int(min % 60)
    if(int(min / 10) == 0):
        minstr = '0' + str(min)
    else:
        minstr = str(min)
    now = datetime.utcnow().strftime('%Y%m%d_') + houstr + minstr
    namnow = nam + '_' + now + '.' + typ
    return namnow
# dateiname, dateiendung, time nachdem neue datei
# minute
# stunde als string
# zeit für dateinamen

```

8 Quellenverzeichnis

Textquellen (mit Datum des letzten Aufrufs):

[1] Quelle: <https://de.wikipedia.org/wiki/Erdmagnetfeld> (14.04.2022)

[2] Quelle: <https://www.ptb.de/cms/nc/ptb/fachabteilungen/abt2/fb-25/ag-251/live-daten-erdmagnetfeldmessung.html> (14.04.2022)

[3] <https://www.itwissen.info/XMR-Sensor-X-magneto-resistive-XMR.html> (17.04.22)

[4] <https://www.digikey.de/de/datasheets/honeywellmicroelectronicsprecisionsensors/honeywell-microelectronics-precision-sensors-hmc5883l> (17.04.22)

[5] https://de.wikipedia.org/wiki/Anisotroper_magneto-resistiver_Effekt (17.04.22)

[6] <https://de.wikipedia.org/wiki/Erdatmosphäre> (19.04.22)

[7] <https://de.wikipedia.org/wiki/Troposphäre> (19.04.22)

[8] <http://radiosonde.eu/RS00-D/RS02H-D.html> (19.04.22)

[9] https://www.dwd.de/DE/wetter/thema_des_tages/2021/4/11.html (19.04.22)

[10] <https://de.wikipedia.org/wiki/Magnetosphäre> (19.04.22)

[11] <https://www.mikrocontroller.net/topic/60869> (22.04.22)

Titelbild:

Abb. 1 - Erdmagnetfeld: istockphoto.com

Verwendete Software:

Spider: Python IDE, Erstellung der Plots

Thonny: Python Editor auf dem Raspi, Erstellung der Programme

GoodNotes: Erstellung der Grafiken