

Spectrometer II

Programmbeschreibung (Community Version)

Stand 7. März 2020

Astropeiler Stockert e.V.

**Astropeiler Stockert 2-4
53902 Bad Münstereifel**

Inhaltsverzeichnis

1	Allgemeines.....	1
2	Funktionsumfang.....	2
2.1	Grundfunktionen.....	2
2.2	Erweiterte Funktionen.....	2
2.3	Unterstützte SDRs.....	2
3	Konfiguration und Verwendung	4
3.1	Konfigurationsdatei	4
3.2	Erste Verwendung.....	6
3.3	Kommandozeilenparameter	6
4	Ausgabeformate und -Dateien	10
4.1	Dateinamen	10
4.2	FITS-Dateien	10
4.3	CSV-Dateien.....	11
5	Beispiele.....	12
6	Verwendung von „CLASS“ für die Auswertung.....	16
7	Image für Raspberry Pi.....	17
8	Rechtliches.....	18

1 Allgemeines

Spectrometer_II ist ein Programm für Messung/Beobachtung von radioastronomischen Spektrallinien. Es unterstützt eine große Anzahl von Software Defined Radios (SDR) und basiert darauf, dass die komplexen Datensamples, die vom SDR geliefert werden, Fouriertransformiert werden.

Die so erhaltenen Spektren werden für die weitere Verarbeitung abgespeichert bzw. optional auch direkt dargestellt.

Das Programm basiert auf „SoapySDR“ⁱ für die Ansteuerung des SDR und „soapy_power“ⁱⁱ für die Fouriertransformation. Es ist in Python (Version 3) geschrieben und erfordert die Installation verschiedener Python Bibliotheken. Es ist für das Betriebssystem Linux ausgelegt und wurde unter Debian 10 entwickelt. Das Programm ist kommandozeilengesteuert. Es gibt keine grafische Oberfläche. Die Kommandosprache und die Programmierungen sind in englischer Sprache, um es auch in einer internationalen Community verwenden zu können.

Um Neueinsteigern die zum Teil komplexe Installation der erforderlichen Pakete zu ersparen, wird ein Image für einen Raspberry Pi bereitgestellt. Dieses enthält dann ein voll funktionsfähiges System und zusätzlich das Programmpaket CLASSⁱⁱⁱ für die Auswertung.

Das Programm wurde für die Verwendung am Observatorium „Astropeiler Stockert“ entwickelt. Die Community-Version enthält nicht die Programmkomponenten für die direkte Kommunikation mit der Teleskopsteuerung, da diese nicht allgemein verwendbar ist.

WICHTIG: Der Autor und der Verein werden keinen Support für dieses Programm leisten. Anfragen zu Fehlern, Bedienung, Nutzung etc. werden in der Regel nicht beantwortet werden können. Wir hoffen, dass dieser Support von der Community selber geleistet werden kann.

Theoretisch ist es denkbar, dass dieses Programm auch unter Windows lauffähig ist. SoapySDR gibt es auch für Windows, soapy_power ist ebenso wie dieses Programm in Python geschrieben. Der Teufel mag im Detail stecken. Der Autor und der Verein haben jedoch nicht die Absicht, dies selber zu testen.

2 Funktionsumfang

2.1 Grundfunktionen

- Erfassung von Spektren mit wählbarer Bandbreite (im Rahmen der Fähigkeiten des verwendeten SDR), Auflösung und Integrationszeit
- Abspeichern der Spektren im FITS Format. Der Dateiname wird dabei automatisch generiert. Er enthält immer eine sogenannte „Scannummer“, die fortlaufend vergeben wird.
- Darstellung der Spektren in einem Plot (skalierbar, abspeicherbar)

2.2 Erweiterte Funktionen

- Errechnung von Rektaszension, Deklination, gal. Länge und Breite aus Teleskopstandort und Ausrichtung (Azimut und Elevation) des Teleskops
- Errechnung und Einbeziehung der Korrektur für den Local Standard of Rest
- Berücksichtigung von Systemen mit Konversion auf eine ZF
- Verschieben des Nullpunktes (Ruhefrequenz) um zu vermeiden, dass dieser mit einer störenden zentralen Linie des SDR zusammenfällt
- Abspeichern im csv-Format
- Eingabe von Objektnamen und Kommentaren für abgespeicherten Spektren
- Unterstützung von weiteren Linien neben der Wasserstofflinie
 - Radio-Rekombinationslinien des Wasserstoffs
 - OH-Maserlinien
 - Methanolmaser-Linien
 - Wassermaser-Linien

2.3 Unterstützte SDRs

Es wird unterschieden zwischen solchen SDRs, bei denen die Funktionsfähigkeit des Programms mit tatsächlicher Hardware getestet wurde, und solchen, die grundsätzlich funktionsfähig sein sollten.

Getestete SDRs

- RTL-SDR
- ADALM-Pluto
- Hack-RF
- SDRplay pro
- Lime SDR
- Ettus Research N200 mit WBX und SBX

Unterstützte, aber nicht getestete SDRs

- Airspy
- BladeRF
- Lime SDR mini
- SDRs mit Osmosdr Unterstützung
- RFSpace
- Weitere Ettus-Produkte mit UHD

3 Konfiguration und Verwendung

Das Programm wird durch eine Konfigurationsdatei und durch Kommandozeilenparameter gesteuert. Vor einer ersten Verwendung sollte die Konfigurationsdatei auf die eigenen Bedürfnisse angepasst werden.

3.1 Konfigurationsdatei

Wesentliche Parameter, die seltener verändert werden, sind in einer Konfigurationsdatei enthalten. Diese Konfigurationsdatei heißt (als Default) `observatory.config`. Sie enthält sogenannte „Sections“ für den Teleskopstandort, die Frontendeigenschaften, und den SDR. Diese sind nachstehend beschrieben:

[Observatory]

Name=	hier kommt ein frei gewählter Name hin
Telescope=	auch hier kommt ein frei gewählter Name hin
Lat=	Die geographische Breite des Teleskops (dezimale Schreibweise)
Lon=	Die geographische Länge des Teleskops (dezimale Schreibweise)
Alt=	Die Höhe des Teleskops über NN in Metern

Die Standortparameter werden für die Errechnung von astronomischen Koordinaten, der LSR Korrektur sowie für die Bestimmung der lokalen Sternzeit verwendet. Der Teleskopname erscheint im Dateinamen der einzelnen Spektren.

[Frontend]

Name=	hier kommt ein frei gewählter Name hin
LocalOsc=	Wenn eine Umsetzung auf eine ZF erfolgt, ist hier die Frequenz des LO in MHz anzugeben. Dabei wird angenommen, dass immer ein Abwärtsmischen erfolgt.

Die Section „Frontend“ ist optional. Der Frontendname wird, wenn vorhanden, in den Header der Datei mit den Spektren eingetragen.

[SDR]

Type=	hier kommt ein frei gewählter Name hin.
-------	---

- Gain=** Der Default-Gain für den SDR in dB. Dieser Parameter kann durch Kommandozeilen-Parameter übersteuert werden.
- SampleRate=** Die Default-Samplerate für den SDR in MHz. Dieser Parameter kann durch Kommandozeilen-Parameter übersteuert werden.
- Bins=** Die Default-Anzahl der spektralen Kanäle. Dieser Parameter kann durch Kommandozeilen-Parameter übersteuert werden. Bins bestimmt zusammen mit der SampleRate die Auflösung.
- IntegrationTime=** Die Default-Integrationszeit in Sekunden. Dieser Parameter kann durch Kommandozeilen-Parameter übersteuert werden.
- CalibrationFactor=** Mit diesen Faktor werden die gemessenen Intensitäten multipliziert. Dadurch können ggf. kalibrierte Werte ausgegeben werden.
- FrequencyCorrection=** Mit diesen Faktor werden die Frequenzen multipliziert. Dadurch können Ungenauigkeiten des SDR ausgeglichen werden.
- AntennaSelection=** Manche SDR erfordern die Angabe einer „Antenne“. Dies dient z.B. der Auswahl bei mehreren Eingängen.

Die Angaben zu CalibrationFactor, FrequencyCorrection und AntennaSelection sind optional. Die anderen müssen vorhanden sein.

Es können mehrere SDR Sections vorhanden sein. Diese weiteren müssen jedoch anders benannt werden, z.B. SDR1 usw. Aktiv und gültig für das Programm ist jedoch immer nur die Section SDR.

Eine Beispiel-Konfigurationsdatei ist vorhanden, diese kann auch für Versuche verwendet werden.

Bei Bedarf können mehrere Konfigurationsdateien vorhanden sein und jeweils beim Programmaufruf über einen Kommandozeilenparameter (-config) ausgewählt werden.

3.2 Erste Verwendung

Für einen ersten Test kann das Programm ganz ohne Parameter aufgerufen werden, da alle Parameter optional sind.:

Spectrometer_II

Es empfiehlt sich jedoch, auch den Parameter -plot anzugeben, um das Ergebnis direkt sehen zu können.

3.3 Kommandozeilenparameter

Die verfügbaren optionalen Kommandozeilenparameter können durch Aufruf des Programms mit -h abgerufen werden:

Spectrometer_II -h

Es erfolgt die Ausgabe aller Parameter mit einer Hilfefunktion dazu. Die Parameter, ihre Bedeutung und Verwendung sind nachstehend beschrieben. Der vollständige Name des Parameters muss nicht zwingend ausgeschrieben werden. Es reicht die Eingabe so vieler Buchstaben, dass eine Unterscheidung von anderen Parametern möglich ist.

-time

Dies ist die Integrationszeit in Sekunden. Ohne Angabe dieses Parameters wird der Default-Wert aus der Konfigurationsdatei genommen. Der Parameter beschreibt die tatsächliche Integrationszeit. Nicht enthalten sind die Zeit bis zum Start der Messung sowie die Zeit, die durch die Errechnung der Fouriertransformation verloren geht. Die tatsächliche Dauer der Messung wird daher immer länger als -time sein und hängt u.a. von der Leistungsfähigkeit des Rechners ab.

-sample

Dies ist die Samplerate in MHz. Ohne Angabe dieses Parameters wird der Default-Wert aus der Konfigurationsdatei genommen. Dieser Parameter

bestimmt die Bandbreite der Messung. Die Samplerate darf die maximale Samplerate des verwendeten SDR nicht überschreiten, ansonsten wird ein Fehler angezeigt. Da viele SDRs nicht beliebige Sampleraten unterstützen, wird die tatsächlich verwendete Samplerate an die verfügbare angepasst. Die tatsächliche Samplerate ist dann in der Spektrumsdatei eingetragen.

-bins

Dies ist die Anzahl der spektralen Kanäle. Ohne Angabe dieses Parameters wird der Default-Wert aus der Konfigurationsdatei genommen. Dieser Parameter bestimmt zusammen mit der Bandbreite die spektrale Auflösung.

-gain

Dies ist der Gain des SDR. Ohne Angabe dieses Parameters wird der Default-Wert aus der Konfigurationsdatei genommen. Der Gain darf den maximalen Gain des jeweiligen SDR nicht überschreiten, ansonsten wird ein Fehler angezeigt.

-plot ['low,high']

Die Angabe dieses Parameters bewirkt, dass das erhaltene Spektrum direkt geplottet wird. Die X-Achse dieses Plots ist die Geschwindigkeit, errechnet aus der Dopplerverschiebung. Die Y-Achse ist die Intensität in willkürlichen, linearen Einheit.

Entsprechend der Konvention in der Radioastronomie sind blauverschobene Werte mit negativem Vorzeichen, rotverschobene Werte mit positivem Vorzeichen versehen.

Wenn nur -plot angegeben wird, dann wird der gesamte spektrale Bereich geplottet. Möchte man nur einen Teil darstellen, so kann zusätzlich ein Geschwindigkeitsbereich ausgewählt werden. Beispiel: Man möchte den Geschwindigkeitsbereich von -150 bis 200 km/s dargestellt haben. Dann sieht der Parameter so aus:

-plot ' -150,200'

Achtung: Die Geschwindigkeitswerte müssen in Hochkomma gesetzt werden, und hinter dem ersten Hochkomma muss ein Blank stehen. Ansonsten versucht das Programm das als einen zusätzlichen Parameter zu interpretieren, was zu einer Fehlermeldung führt.

Der Plot selber kann aber auch nachher in bestimmte Bereiche gezoomt werden.

-azel 'Azimut,Elevation'

Diese Parameter gibt Richtung der Antenne als Azimut und Elevation an.

Daraus kann das Programm dann zusammen mit der Position des Observatoriums und der Zeit Rektaszension, Deklination sowie galaktische Länge und Breite errechnen. Diese Daten werden in den Header der Spektrumsdatei geschrieben sowie ggf. für die Errechnung der VLSR Korrektur verwendet.

-eq 'RA,Dec'

Wenn Rektaszension und Deklination bekannt sind, kann dies anstelle von Azimut und Elevation eingegeben werden. Azimut und Elevation sowie galaktische Länge und Breite werden dann errechnet.

-gal 'Lon,Lat'

Als weitere Alternative können galaktische Koordinaten angegeben werden. Äquatoriale Koordinaten sowie Azimut und Elevation werden dann errechnet.

-novlsr

Default-mäßig versucht das Programm, eine Korrektur der Daten auf den Local Standard of Rest durchzuführen. Dazu muss der Standort des Teleskops in der Konfigurationsdatei korrekt eingetragen sein. Weiterhin muss einer der drei Parameter -azel, -eq oder -gal angegeben sein. Fehlen diese Parameter, so wird keine LSR Korrektur ausgeführt und eine Warnung ausgegeben. Will man die Korrektur unabhängig davon abschalten, so geschieht das über diesen Parameter -novlsr.

-shift_center

Viele SDRs zeigen in der Mitte des Spektrums eine schmale, intensive Linie oder sonstige Artefakte. Es ist unschön, wenn diese Linie im beobachteten Spektrum liegt. Mit der Angabe dieses Parameters wird die Frequenz so verschoben, dass das Spektrum außerhalb der Mitte zu liegen kommt und die Problematik so vermieden wird.

-line Linename

Ohne Angabe dieses Parameters geht das Programm davon aus, dass die 21-cm Linie des Wasserstoffs beobachtet werden soll. Die entsprechende Ruhfrequenz wird verwendet. Andere unterstützte Linien sind:

- | | |
|------------|---|
| ○ Hydrogen | 21-cm Wasserstoff (default) |
| ○ H165 | H165 α Radio-Rekombinationslinie |
| ○ H166 | H166 α Radio-Rekombinationslinie |
| ○ H167 | H167 α Radio-Rekombinationslinie |
| ○ H168 | H168 α Radio-Rekombinationslinie |
| ○ H169 | H169 α Radio-Rekombinationslinie |

- | | |
|------------|---|
| ○ H170 | H170 α Radio-Rekombinationslinie |
| ○ OH_1612 | OH 1612 MHz |
| ○ OH_1665 | OH 1665 MHz |
| ○ OH_1667 | OH 1667 MHz |
| ○ OH_1720 | OH 1720 MHz |
| ○ Meth_6 | Methanol 6 GHz |
| ○ Meth_12 | Methanol 12 GHz |
| ○ Water_22 | Wassermaser 22 GHz |

-sdr

Überschreibt den von der Konfigurationsdatei vorgegebenen Namen des SDR.

-obj

Name eines Beobachtungsobjektes. Sofern eingegeben, wird dies Bestandteil des Dateinamens der Messdatei.

-comment

Erlaubt die Eingabe eines Kommentars zur Messung. Dieser findet sich dann im Header der Messdatei wieder.

-lo

Frequenz eines lokalen Oszillators in MHz. Wenn eingegeben, überschreibt dies den von der Konfigurationsdatei vorgegebenen Wert. Es wird immer ein Heruntermischen angenommen, d.h. die Beobachtungsfrequenz wird um die Frequenz des LO reduziert.

-csv

Wird dieser Parameter angegeben, so wird zusätzlich zu einer FITS Datei das Spektrum in einer csv-Datei ausgegeben.

-config

Name (und Pfad) einer Konfigurationsdatei, sofern nicht die Default-Datei verwendet werden soll

-v -vv -vvv

Je nach Angabe werden mehr oder weniger Meldungen ausgegeben. Am Anfang macht es vielleicht Sinn, mit vvv anzufangen um zu sehen, was passiert.

4 Ausgabeformate und -Dateien

4.1 Dateinamen

Für jedes Spektrum werden zwei FITS Dateien und optional eine csv Datei angelegt. Die beiden FITS Dateien sind inhaltlich identisch.

Die erste FITS Datei heißt immer „latest.fits“ und enthält die jeweils letzte Messung. Die Datei wird durch jede Messung überschrieben.

Für die zweite FITS Datei wird der Dateiname wie folgt gebildet:

`telescope_object_scanno_spec.fits`

wobei telescope der Teleskopname aus der Konfigurationsdatei, object der angegebene Objektname und scanno die aktuelle Scannummer ist.

Sofern kein Objektname eingegeben ist, steht dort „NONE“.

Die dritte optionale Datei ist eine csv Datei. Der Dateiname entspricht der FITS Datei, jedoch mit der Endung .csv

4.2 FITS-Dateien

FITS steht für Flexible Image Transfer System und ist ein Standard-Datenformat in der Astronomie. Auch wenn das Wort „Image“ benutzt wird, so ist es keinesfalls auf Bilder beschränkt. In der Radioastronomie dient es z.B. ebenfalls für Spektren.

Die von diesem Programm erzeugte FITS Datei enthält im Header zahlreiche Parameter. Im Header selber steht eine Erläuterung, wofür diese Parameter in Einzelnen stehen. Ein Beispiel ist im Anhang zu finden. Die Parameterauswahl ist so getroffen, das auf jeden Fall alle von dem Programm CLASS (Bestandteil von GILDAS) erforderlichen Parameter enthalten sind und die Spektren dort direkt eingelesen werden können.

4.3 CSV-Dateien

Die csv Datei beginnt mit einem Header, der dem Header der FITS Datei entspricht. Dieser wird gefolgt von den Daten in vier Spalten, die jeweils durch ein Komma getrennt sind:

1. Spalte: fortlaufende Kanalnummer
2. Spalte: Frequenz (in MHz)
3. Spalte: Geschwindigkeit (in km/s)
4. Spalte: Intensität (linear)

Frequenz und Geschwindigkeit sind LSR-korrigiert, sofern dies möglich war und nicht abgeschaltet war.

5 Beispiele

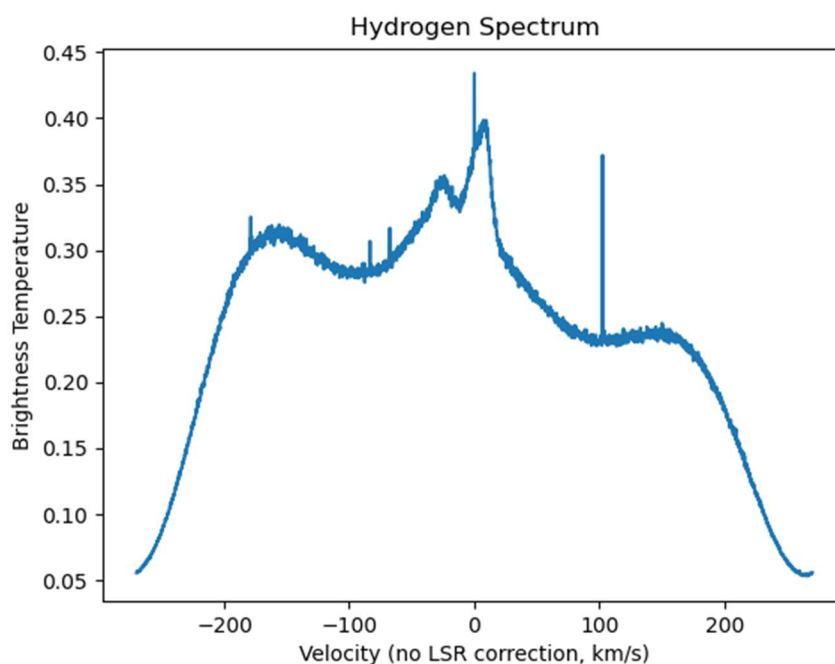
Alle Beispiele wurden am 3-m Teleskop des Astropeiler Stockert mit jeweils 20 s Integrationszeit als Defaultwert aufgenommen. Die in den Spektren auftauchenden schmalen Linien sind übrigens Störungen (RFI).

Das erste Beispiel ist ein Wasserstoffspektrum, bei dem ein RTL-SDR verwendet wurde. Es wurde keine VLSR Korrektur errechnet, da keine Beobachtungsrichtung angegeben war. Die Samplingrate war 2,56 MHz (in der Konfigurationsdatei vorgegeben).

Die Kommandozeile ist dann einfach:

```
Spectrometer_II -plot
```

Der Parameter -plot bewirkt, dass gleich ein Plot mit dem Spektrum ausgegeben wurde:

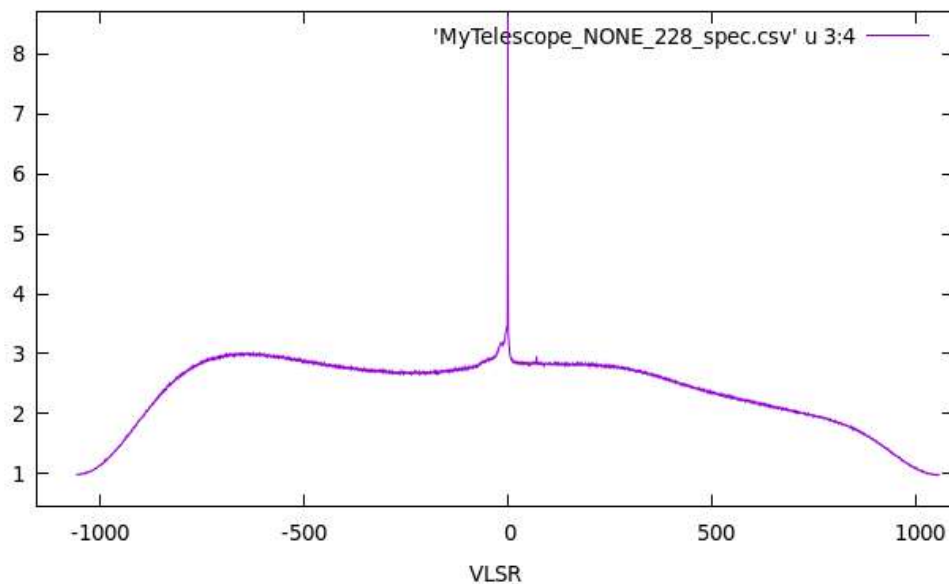


Der RTL-SDR hat eine charakteristische Struktur mit 3 „Buckeln“, die es bei schwachen Signalen schwer machen kann die eigentliche Wasserstofflinie zu erkennen. In diesem Fall ist das Signal jedoch deutlich in der Mitte zu erkennen.

Das nächsten Beispiele sind mit einem Lime-SDR aufgenommen worden. Die Samplingrate betrug 10 MHz und wurde in der Konfigurationsdatei vorgegeben. In diesem Fall wurden die unten gezeigten Plots aus den csv-Dateien mit gnuplot erzeugt. Die Beobachtungsrichtung war $128,7^\circ$ galaktische Länge und $-11,8^\circ$ galaktische Breite, die VLSR Korrektur wurde errechnet.

```
Spectrometer_II -gal `128.7, -11.8`
```

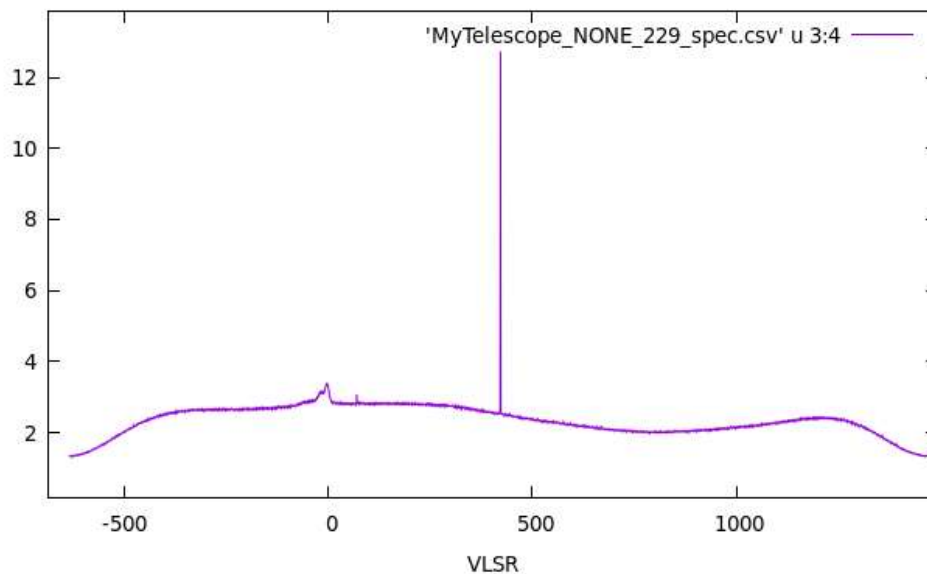
Dies ergibt:



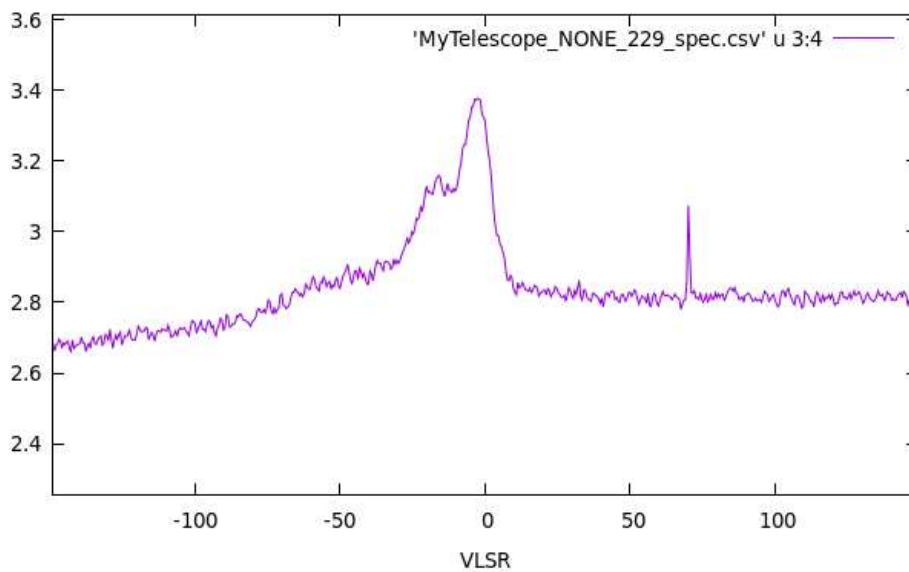
Der Lime-SDR hat wie viele SDRs genau in der Mitte eine starke Störlinie. Dies vermeidet man, indem man die Ruhefrequenz des Wasserstoffts nicht genau in die Mitte legt. Dies macht der Parameter `-shift`:

```
Spectrometer_II -gal `128.7, -11.8` -shift
```

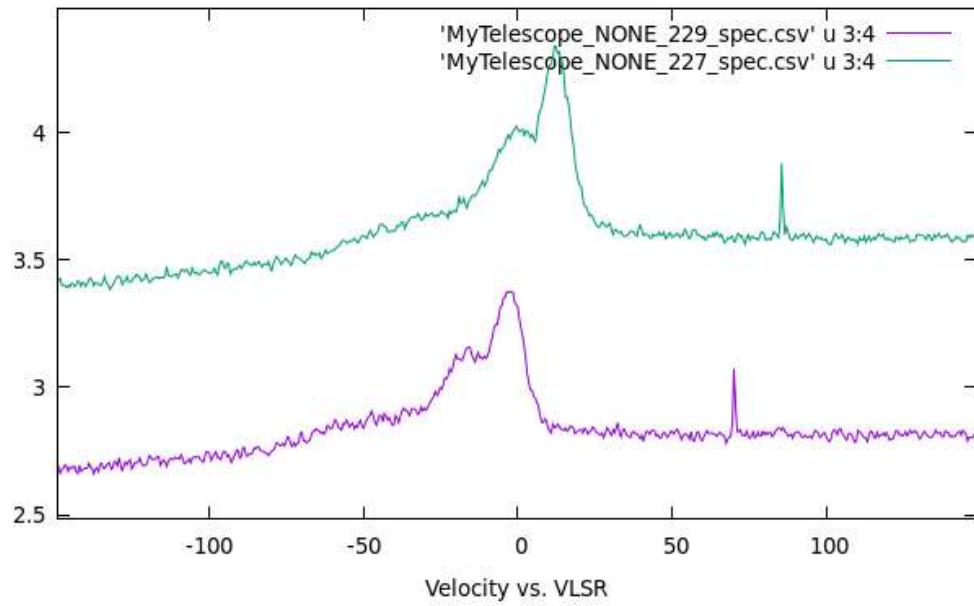
Damit sieht es dann schon besser aus:



Durch die hohe Samplingrate wird ein breiter spektraler Bereich erfasst. Daher ist es sinnvoll, in den interessanten Bereich hinein zu zoomen:



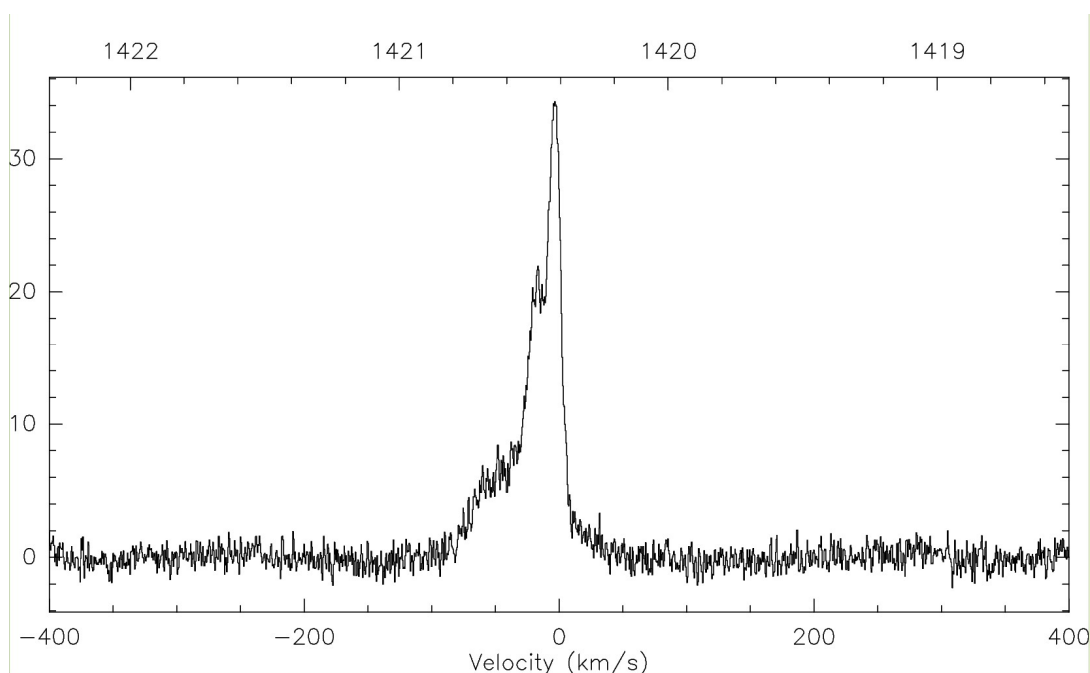
Interessant ist in diesem Fall auch ein Vergleich, wenn man die VLSR Korrektur durch den Parameter -novlsr abschaltet. Hier ein Vergleich:



Der Unterschied beträgt immerhin 15,5 km/s und ist daher nicht vernachlässigbar.

6 Verwendung von „CLASS“ für die Auswertung

Für eine bestmögliche Auswertung ist die Verwendung von CLASS empfohlen. Hier werden FITS Dateien als Eingabe verwendet. Sicher ist eine gewisse Einarbeitung vonnöten, aber es ist lohnend. An dieser Stelle soll kein Tutorial stehen. Als Appetitanreger sei nur das gleiche, vom Lime-SDR aufgenommene Spektrum nach einer Basislinienkorrektur, Kalibrierung und Entfernung der Störlinie gezeigt:



7 Image für Raspberry Pi

Wie unter Allgemeines bereits erwähnt, ist es das Einfachste ein Image für einen Raspberry Pi zu verwenden. Damit umgeht man alle Probleme, die beim Installieren der notwendigen Pakete, die für das Programm „Spectrometer II“ erforderlich sind, auftreten können. Das Image enthält auch eine fertig kompilierte Version von Gildas/Class, so dass diese Auswertungssoftware ebenfalls zur Verfügung steht.

Für die Funktion des Programms ist jedoch ein Raspi der Version 4 mit 4 GByte RAM erforderlich. Mit den Raspis der Version 3 hat sich kein stabiler Betrieb ergeben. Ob ein Raspi 4 mit 2 GByte ausreichend ist, konnte nicht ausprobiert werden.

8 Rechtliches

Das Urheberrecht (Copyright) für dieses Programmes liegt bei Wolfgang Herrmann, Astropeiler Stockert e.V.

Es ist gestattet, dieses Programm zu nutzen, kopieren, weiterzugeben, zu verändern und weiter zu entwickeln. Diese Erlaubnis gilt solange dies ausschließlich für persönliche und wissenschaftliche Zwecke erfolgt. Nicht gestattet ist die Verwendung für kommerzielle Zwecke jeder Art, insbesondere der Verkauf und das Bundling mit einem kommerziellen Produkt.

Der Autor übernimmt keinerlei Gewährleistung und Haftung für dieses Programm. Es wird „as is“ zur Verfügung gestellt und die Verwendung liegt ausschließlich in der Verantwortung des Nutzers.

ⁱ <https://github.com/pothosware/SoapySDR/wiki>

ⁱⁱ https://github.com/xmikos/soapy_power

ⁱⁱⁱ <https://www.iram.fr/IRAMFR/GILDAS/>