BCCM BLOCKCHAIN COMPETENCE CENTER MITTWEIDA

Blockstream Satellite – die Bitcoin-Blockchain per Satellit

Technischer Report, Version vom 02.03.2018

Autoren: Volker Delport, Vinzenz Lorenz, Timur Karychev

Inhalt

1. Einführung und Motivation 2
2. Satelliten 2
3. Satellitenempfangsanlage 3
3.1. Azimut, Elevation und Skew 3
3.2. Satellitenspiegel 5
3.3. Low Noise Block Converter 5
3.4. Ausrichtung der Satellitenantenne 6
3.5. Empfänger-Hardware 7
3.5.1. Preiswerter Aufbau 7
3.5.2. Empfangsanlage im Labor 8
3.6. SDR-Empfänger 9
3.6.1. SDR-Empfänger-Hardware 9
3.6.2. SDR-Empfänger-Software 10
4. Inbetriebnahme und Test 12
4.1. Software-Installation 12
4.2. Aktualisierung der Bitcoin-Blockchain über eine Internetverbindung 13
4.3. Aktualisierung der Bitcoin-Blockchain über Satellit 14
4.3.1. Empfang des Blockstream-Satellitensignals mit rx_gui.py 14
4.3.2. Aktualisierung der Bitcoin-Blockchain mit rx.py 16
5. Fazit 19
6. Referenzen 21

1. Einführung und Motivation

Am 15. August 2017 verkündete Dr. Adam Back, Mitbegründer und CEO des Startup-Unternehmens Blockstream Inc., offiziell den ersten Satellitendienst für Bitcoin [1].

Blockstream hat Transponder von momentan drei geostationären Satelliten gemietet, worüber die Bitcoin Blockchain bereits auf vier Kontinente - Afrika, Europa, Nord- und Südamerika - verteilt wird. Damit erreicht die Bitcoin auch Regionen der Erde, in denen kein Internetzugang vorhanden ist.

Allerdings muss man sich klar machen, dass die Satellitenverbindung nur eine Richtung kennt. Das heißt, man kann eine durchgeführte Transaktion durch Blockstream Satellite lediglich überprüfen. Senden muss man die Transaktion, die weniger als ein Kilobyte in Anspruch nimmt, über ein anderes Kommunikationsmittel, z. B. über eine Satellitentelefonverbindung.

In den FAQs des Projekts gibt Blockstream an, dass für die Satellitenverbindung eine Netto-Datenrate von 64 kbit/s garantiert wird [2]. Mit dieser Datenrate würde man ca. 230 Tage benötigen, um einen Bitcoin-Knoten auf die ca. 159 Gigabyte[1] große Bitcoin-Blockchain zu synchronisieren. Aus diesem Grund strahlt Blockstream Satellite nur die Blöcke der letzten 24 Stunden aus. Der geneigte Nutzer benötigt eine fast zu 100 Prozent gespeicherte Bitcoin-Blockchain, bevor er sich auf den Weg in die Sahara macht.

Ein mit dem Internet verwöhnter Europäer käme nicht auf die Idee, seine Bitcoin-Transaktionen über Blockstream Satellite zu überprüfen. Für uns war das Projekt aber aus technischer Sicht interessant, da Blockstream eine Empfängerlösung auf der Basis eines Software Defined Radio (SDR) entwickelt hat, die es ermöglicht, einen Blockstream-Satellitenempfänger preiswert aufzubauen.

Mit diesem Beitrag wollen wir unsere Erfahrungen beim Aufbau des Blockstream-Satellitenempfängers weitergeben und gleichsam dem interessierten Leser ein Manual zur Verfügung stellen, mit dem sich die einzelnen Installationsschritte nachvollziehen lassen sollten.

2. Satelliten

Blockstream verteilt die Bitcoin-Blockchain momentan über drei geostationäre Satelliten, Galaxy 18 (Nordamerika), Eutelsat 113 (Südamerika) und Telstar 11N (Afrika und Europa).

Der Telstar 11N 37,5° W ist ein kanadischer, geostationärer Telekommunikationssatellit. Er wurde am 26. Februar 2009 mit einer russischen Zenit-3-SLB-Rakete vom Kosmodrom Baikonur (Kasachstan) in den Transferorbit eingeschossen. Mithilfe seiner eigenen Antriebswerke setzte der ca. 4 Tonnen schwere Satellit seine Reise fort, bis er die gewünschte Position im geostationären Orbit einnahm. Der Telstar 11N trägt 39 Ku-Band-Transponder, mit denen er Empfangsanlagen in Nordamerika, Afrika und Westeuropa erreicht. Außerdem strahlt der Satellit auf den Atlantik, um Schiffe und Flugzeuge mit Breitbandkommunikation zu versorgen. Seine Betriebsdauer wird auf 15 Jahre geschätzt.

[1] am 02.03.2018, Quelle [10]

Das Ku-Band ist für die Satellitenkommunikation in zwei Frequenzbänder eingeteilt. Die Downlink-Frequenzen (Satellit zu Empfangsanlagen) reichen im unteren Frequenzband (Low) von 10,71 bis 11,69 GHz und im oberen Frequenzband (High) von 11,72 bis 12,75 GHz. Die beiden Frequenzbänder sind in kleinere Frequenzkanäle unterteilt. In jedem Frequenzkanal arbeitet ein Transponder, der wie ein Repeater das Signal von der Erde auf der Uplink-Frequenz empfängt, das Signal verstärkt, auf die Downlink-Frequenz umsetzt und wieder auf die Erde abstrahlt.

Um Frequenzbandbreite einzusparen, werden die Signale neben der Frequenz auch in der Polarisation unterschieden. Unter der Polarisation versteht man in der Funktechnik die Ausrichtung des elektrischen Feldstärkevektors der elektromagnetischen Welle gegenüber ihrer Ausbreitungsrichtung. In dem betreffenden Ku-Band unterscheidet man zwischen horizontaler und vertikaler Polarisation. Bei idealer Depolarisation könnte man das horizontal polarisierte Signal und das vertikal polarisierte Signal auf demselben Frequenzkanal senden und der Empfänger könnte trotzdem beide Signale voneinander trennen. Auf dem langen Weg zwischen Satellit und Erde wird die Polarisation der Satellitensignale jedoch insbesondere durch das Erdmagnetfeld gedreht, so dass in der Praxis jeweils nur ca. die Hälfte der Kanalbandbreite eingespart werden kann.

Das Blockstream-Signal sendet der Telstar 11N 37,5° W im unteren Frequenzband auf 11.504,02 MHz mit vertikaler Polarisation.

3. Satellitenempfangsanlage

3.1. Azimut, Elevation und Skew

Telekommunikationssatelliten, die sich auf einer geostationären Umlaufbahn bewegen, haben den Vorteil, dass ihre Signale mit einer stationären Antennenanlage empfangen werden können. Damit ein Satellit geostationär ist, muss er sich auf einer Kreisbahn in der Äquatorebene mit der Erde (Drehrichtung Osten) bewegen, wobei seine Umlaufzeit der Rotationsdauer der Erde entsprechen muss. Diese Umlaufzeit wird erreicht, wenn die Umlaufbahn des Satelliten ca. 35.786 km von der Äquatoroberfläche entfernt ist.

Um die Signale eines geostationären Satelliten an einem bestimmten Ort der Erde empfangen zu können, muss die Satellitenantenne auf den Subsatellitenpunkt ausgerichtet werden, den der Satellit auf den Äquator setzt. Die Winkel für die horizontale (Azimut) und die vertikale Ausrichtung (Elevation) der Antenne werden somit durch den Subsatellitenpunkt sowie durch den Längen- und Breitengrad des Empfangsorts bestimmt.

Mit den Gleichungen (1) und (2) können Azimut bzw. Elevation bestimmt werden[2].

Im Gegensatz zu den in Deutschland bekannten TV-Satelliten, wie z. B. Astra 19,2° O oder Eutelsat 13,0° O, liegt der Subsatellitenpunkt des Telstar 11N 37,5° W deutlich westlich vom Nullmeridian. Der Azimut, mit dem die Satellitenantenne auf den Telstar 11N 37,5° W in Mittweida (Länge: 12,98° O, Breite: 50,98° N) ausgerichtet werden muss, beträgt ca. 237,3 °, d. h. die Antenne muss in Mittweida aus Südrichtung (180°) um ca. 57 ° in Richtung Westen gedreht werden. Die Elevation der Satellitenantenne beträgt für Mittweida ca. 15,3 °.

[2] Die Längen- und Breitengrade sind in Grad einzusetzen. Die Längengrade westlich des Nullmeridians und die Breitengrade südlich des Äquators haben negative Vorzeichen.

$$\Delta L = L_E - L_S$$

$$\tan(\alpha) = \frac{-\tan(\Delta L)}{\sin(B_E)}$$

$$Az = \begin{cases} \alpha & B_E < 0, \Delta L < 0 \\ 360° - \alpha & B_E < 0, \Delta L > 0 \\ 180° + \alpha & B_E > 0, \Delta L < 0 \\ 180° - \alpha & B_E > 0, \Delta L > 0 \end{cases} \quad (1)$$

L_E: $Längengrad\ Empfangsort$
L_S: $Längengrad\ Subsatellitenpunkt$
B_E: $Breitengrad\ Empfangsort$

$$\Delta L = L_E - L_S$$

$$K = cos(B_E) \cdot cos(\Delta L)$$

$$El = arctan\left(\frac{K - \frac{R}{R+h}}{\sqrt{1-K^2}}\right) \quad (2)$$

R: $Erdradius\ am\ Äquator$: $6.378\ km$
h: $Höhe\ geostationärer\ Orbit$: $35.786\ km$

Neben dem Azimut- und dem Elevationswinkel der Satellitenantenne ist der Polarisationswinkel (LNB-Kippwinkel, engl. Skew) der Antenne von Bedeutung. Der Skew gibt die Drehung der vertikalen Polarisationsebene des Satellitenempfangssignals von der geografischen Vertikalen an. Um eine Depolarisation des Empfangssignals und damit seine Abschwächung zu vermeiden, sollte der LNB mit dem Skew aus dem Lot gedreht werden. Der Skew kann mit Hilfe der Gleichung (3) bestimmt werden.

$$Skew\ [°] = -\arctan\left(\frac{\sin(L_S - L_E)}{\tan(B_E)}\right) \quad (3)$$

L_E: $Längengrad\ Empfangsort$
L_S: $Längengrad\ Subsatellitenpunkt$
B_E: $Breitengrad\ Empfangsort$

Der Skew für den Empfang des Telstar 11N 37,5° W in Mittweida beträgt ca. +32 °.

Der Praktiker verlässt sich i. A. auf im Internet verfügbare Listen, Online-Rechner oder Apps, in denen die o. g. Gleichungen umgesetzt sind.

3.2. Satellitenspiegel

Die in unseren Breiten einzustellende Elevation der Satellitenantenne ist sehr niedrig. Sie beträgt für Mittweida ca. 15,3 °. Im Vergleich dazu beträgt die Elevation für den Empfang von Astra 19,2° O in Mittweida ca. 31,3 °. Eine geringe Elevation ist nachteilig, da der Satellitenspiegel in Richtung Erdoberfläche geneigt werden muss und somit das Satellitensignal durch das Wärmerauschen der Erde zunehmend störend beeinflusst wird. Je schlechter das Verhältnis zwischen der Signalleistung und der Rausch- bzw. Störleistung ist, desto schwieriger wird es, den digitalen Datenstrom des Satelliten im Empfänger korrekt zu detektieren.

Für den Empfang des Telstar 11N 37,5° W in Deutschland empfehlen wir deshalb einen Satellitenspiegel mit einem Durchmesser von 120 cm einzusetzen.

3.3. Low Noise Block Converter

Der Satellitenspiegel ist lediglich für die Bündelung der vom Satelliten ausgestrahlten, hochfrequenten elektromagnetischen Wellen verantwortlich. Die Aufnahme der Mikrowellen, ihre rauscharme Verstärkung und ihre Umsetzung in das sogenannte SAT-Zwischenfrequenzband erledigt der Low Noise Block Converter (LNB).

Das auf den Satellitenspiegel gerichtete, konische Feed-Horn des LNB fängt die Mikrowellen auf, die danach über ein kurzes Hohlleiterstück zu zwei Antennen gelangen. Die beiden Antennen sind auf die Platine des LNB gelötet und ragen in den Hohleiter hinein. Beide Antennen sind senkrecht zueinander angeordnet, so dass sie jeweils den horizontal bzw. vertikal polarisierten Anteil der elektrischen Feldstärke auffangen und in elektrische Signale umwandeln. Jede Antenne speist das horizontal bzw. vertikal polarisierte Signal in separate Signalpfade, in denen die weitere Signalverarbeitung symmetrisch erfolgt. Die Signalempfangsleistung wird jeweils durch einen Verstärker angehoben, an den hohe Anforderungen gestellt werden. Auf der einen Seite muss der Verstärker ein möglichst geringes Rauschmaß besitzen, da er nach der Antenne die erste Komponente des Satellitenempfängers ist und sein Eigenrauschen das Signal-Rauschleistungsverhältnis des Satellitenempfängers direkt beeinflusst. Auf der anderen Seite muss der Verstärker eine möglichst hohe Verstärkung aufweisen, wodurch der Einfluss des Eigenrauschens der nachfolgenden Systemkomponenten fast vollständig unterdrückt werden kann. Nach der rauscharmen Verstärkung werden die hochfrequenten Signale in die niedrigere, sogenannte SAT-Zwischenfrequenz (950 MHz – 2.150 MHz) durch die Mischung mit der harmonischen Schwingung eines Lokaloszillators umgesetzt. Die Frequenzumsetzung reduziert die Signalleistungsdämpfung der nachfolgenden Systemkomponenten, insbesondere des Koaxialkabels zwischen LNB und Satellitenempfänger. Die Wahl der Oszillatorfrequenz bestimmt, in welches Frequenzband die Signale umgesetzt werden. Für das untere Frequenzband (Low-Band) wurde eine Oszillatorfrequenz von 9,75 GHz und für das obere Frequenzband (High-Band) von 10,60 GHz festgelegt. Mit Hilfe der Gleichung (4) kann die jeweilige SAT-Zwischenfrequenz bestimmt werden.

$$f_{ZF} = f_{HF} - f_O \tag{4}$$

f_{ZF}: $Zwischenfrequenz$
f_{HF}: $Hochfrequenz$
f_O: $Oszillatorfrequenz$

Um das gewünschte Signal zu empfangen, muss der Satellitenempfänger dem LNB die Polarisationsebene und die Frequenzbandlage mitteilen. Dies geschieht durch Anlegen einer Gleichspannung (+18V: horizontal, +14V: vertikal) sowie durch Ein- und Ausschalten eines 22-kHz-Signals (aus: unteres Frequenzband, ein: oberes Frequenzband).

Bei der Auswahl eines geeigneten LNB für den Empfang des Blockstream-Satellitensignals muss auf die Frequenzstabilität bei der Erzeugung der Lokaloszillatorschwingung im LNB geachtet werden. Die meisten, marktüblichen LNB sind für den breitbandigen DVB-S/S2-Empfang in Deutschland ausgelegt und besitzen nicht die gewünschte Frequenzstabilität, die bei der Umsetzung des schmalbandigen Blockstream-Satellitensignals erforderlich ist. Ein PLL-LNB, d. h. ein LNB, dessen Frequenzstabilität durch eine integrierte Phasenregelung (PLL, Phase-Locked Loop) verbessert wurde, erfüllt diese Bedingung. Glücklicherweise sind PLL-LNB für das Ku-Band als Single-, Twin-, Quad- und Quattro-LNB in Deutschland einfach erhältlich und auch nicht teurer als die herkömmlichen TV-LNB. Single-, Twin- und Quad-LNB unterscheiden sich in der Anzahl der anschließbaren Satellitenempfänger. Wenn mit der Anlage nur das Blockstream-Signal empfangen werden soll, genügt ein Single-LNB.

3.4. Ausrichtung der Satellitenantenne

Bei der Ausrichtung einer Satellitenantenne sollte man prinzipiell folgendermaßen vorgehen:

Der LNB wird in die vorgesehene Halterung gesetzt und auf den gewünschten Skew gedreht. Steht man hinter dem Spiegel und schaut in die Richtung des Satelliten, muss der LNB bei einem positiven Skew im Uhrzeigersinn bzw. bei einem negativen Skew gegen den Uhrzeigersinn gedreht werden. Zunächst wird die Elevation des Spiegels grob anhand der Markierungen an der Antennenhalterung eingestellt. Mit geeigneten Geräten werden die Empfangsfrequenz, die Frequenzlage (Low/High) und die Polarisationsebene (horizontal/vertikal) des gewünschten Signals vorgegeben. Dazu kann z. B. ein Satelliten-Receiver in Kombination mit einem SAT-Finder oder ein Messempfänger eingesetzt werden. Anschließend wird die Antenne in Richtung Süden (180°) ausgerichtet und je nach Azimut solange nach Osten oder Westen gedreht, bis die Signalempfangsstärke maximal wird. Abschließend erfolgt noch einmal die Feinjustierung der Elevation, wodurch die bestmögliche Signalempfangsleistung erzielt wird.

Der Telstar 11N 37,5° W gehört aufgrund seiner Lage nicht zu den in Deutschland normalerweise genutzten Satelliten. Während des Vorhabens strahlte der Satellit kein DVB-S/S2-Signal über seinen europäischen Beam ab. Um eine erste Orientierung zu erhalten, empfehlen wir deshalb, die Satellitenantenne zunächst auf den Hispasat 30° W auszurichten, da dieser in der Nähe des Telstar 11N 37,5° W liegt und unverschlüsselte DVB-S-Signale ausstrahlt. Sobald ein ausgewählter Fernsehsender des Hispasat 30° W empfangen wurde, kann man den Spiegel auf den Telstar 11N 37,5° W weiterdrehen. Zuvor müssen natürlich die Empfangsfrequenz auf 11.504,02 MHz, der Skew und die Elevation neu eingestellt werden. Der anschließende Abstimmprozess erfolgt wie oben beschrieben.

Auf dem Dach des Grunert-de-Jacomé-Baus (Haus 6) ist ein 120-cm-Satellitenspiegel installiert, den wir für unser Vorhaben auf den Telstar 11N 37,5 ° W ausgerichtet haben (Abbildung 1).





Abbildung 1: auf den Telstar 11N 37,5 ° W ausgerichtete Satellitenantenne

Für die Ausrichtung der Satellitenantenne nutzten wir einen Messempfänger. Mit dem Messempfänger kann man dem LNB das Frequenzband und die Polarisation des gewünschten Satellitensignals mitteilen sowie die Empfangsfrequenz einstellen. Der Messempfänger zeigt die Signalempfangsstärke an und gibt außerdem ein akustisches Signal aus, an dessen Tonfrequenz sich die Änderung der Signalempfangsstärke abschätzen lässt.

3.5. Empfänger-Hardware

3.5.1. Preiswerter Aufbau

Die Abbildung 2 zeigt den schematischen Aufbau einer preiswerten Blockstream-Satellitenempfangsanlage.

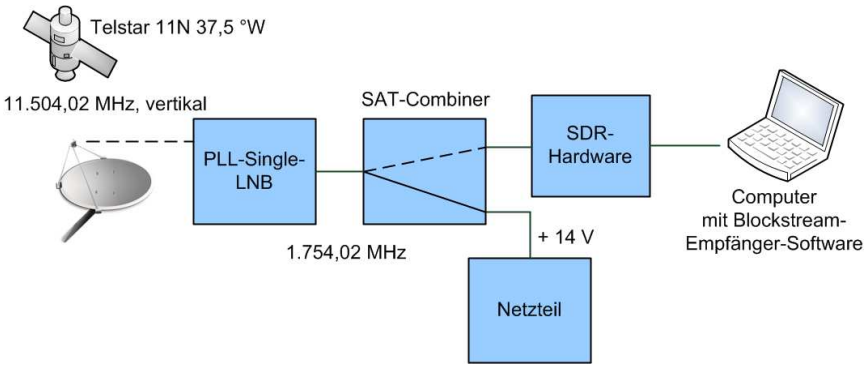


Abbildung 2: Einfache Blockstream-Satellitenempfangsanlage

Ein PLL-Single-LNB wird über ein Koaxialkabel mit einer passiven Frequenzweiche (SAT-Combiner) verbunden, die für den SAT-Zwischenfrequenzbereich (950 MHz – 2.150 MHz) ausgelegt ist. Bei der Auswahl der Frequenzweiche ist zu beachten, dass sie nur über einen der beiden Pfade eine Gleichspannung (DC) passieren lässt.

An diesen Pfad schließt man ein Netzteil mit F-Stecker an, welches eine +14V-Gleichspannung erzeugt, so dass der LNB die Signale in der vertikalen Polarisationsebene liefert. Die Gleichspannung wird über den F-Stecker zwischen Innen- und Außenleiter des Koaxialkabels angelegt. Damit der SDR-Empfänger durch die +14V-Gleichspannung keinen Schaden nimmt, verbindet man ihn mit dem anderen Pfad der Frequenzweiche, d. h. mit dem Pfad ohne den DC-Durchlass. Je nachdem welchen Steckverbindungstyp der SDR-Empfänger besitzt, benötigt man noch einen Adapter zum F-Stecker am Koaxialkabel. Mit seinem USB-Stecker wird SDR-Empfänger mit einem Notebook verbunden, auf dem die SDR-Software läuft. Da durch den Combiner und die Steckverbindungen bereits Signal-empfangsleistung verloren geht, sollten die Koaxialkabel möglichst kurz ausfallen, um die Signaldämpfungen in Grenzen zu halten. Das Blockstream-Satellitensignal befindet sich im unteren Frequenzband. Man braucht also nicht einmal ein 22-kHz-Signal.

3.5.2. Empfangsanlage im Labor

In unserem Labor befindet sich bereits eine Satellitenempfangsanlage mit Multischalter für mehrere Teilnehmer, die wir nun auch für den Empfang des Blockstream-Satelliten nutzen.

Dazu setzen wir statt des Single-LNB einen Quattro-LNB ein, der an seinen vier Ausgängen jeweils eine der vier Signalkombinationen aus Frequenzband (L: low, H: high) und Polarisation (H: horizontal, V: vertikal), d. h. L/H, L/V, H/H und H/V, liefert. Die vier Ausgänge des PLL-Quattro-LNB werden über einen Verstärker, der sich auf dem Dach in unmittelbarer Nähe der Satellitenantenne befindet, zu einem Multischalter im Labor geführt. Am anderen Ende des Multischalters können mehrere Satelliten-Receiver angeschlossen werden. Für unseren Blockstream-Satellitenempfänger verbinden wir den Multischalter und den SDR-Empfänger über ein DiSEqC-Gerät[3], mit dem wir bequem das Frequenzband und die Polarisation per Taster umschalten können. In unserem Fall besitzt der SDR-Empfänger am Antenneneingang eine SMA-Buchse, so dass ein Adapter von F-Stecker auf SMA-Buchse notwendig ist. Mit seinem USB-Stecker verbinden wir schließlich den SDR-Empfänger mit einem Notebook (Abbildung 3).



Abbildung 3: Multischalter (links) und Satellitenempfänger (rechts) im Labor

[3] Digital Satellite Equipment Control (DiSEqC) ist ein offener Industriestandard zur Steuerung von Satellitenempfangsanlagen.

3.6. SDR-Empfänger

Das Ziel von Software Defined Radio (SDR) beschreibt Eric Blossom, der Gründer des Open-Source-Projekts GNU Radio, treffend in einem Satz: „Bringing the code as close to the antenna as possible is the goal of software radio.“ [3]

Ein Software Defined Radio (SDR) ist ein Sende- und/oder Empfangssystem zur drahtlosen Informationsübertragung, bei dem Teile der Signalverarbeitung durch Software realisiert werden. Bezogen auf den SDR-Empfänger besteht die Grundidee von Software Defined Radio darin, das Funksignal möglichst früh nach der Empfangsantenne zu digitalisieren und die anschließende Signalverarbeitung mit Hilfe einer Software durchzuführen. SDR-Sender bzw. -Empfänger haben gegenüber einer analogen Hardware den Vorteil, dass man sie konfigurieren und programmieren und somit flexibel und schnell auf geänderte Spezifikationen oder neue Standards anpassen kann.

3.6.1. SDR-Empfänger-Hardware

Die SDR-Empfänger-Software von Blockstream Satellite ist kompatibel für SDR-Hardware-Module der RTL-SDR-Serie. Die Abbildung 4 zeigt das stark vereinfachte Prinzip einer SDR-Empfängervariante.

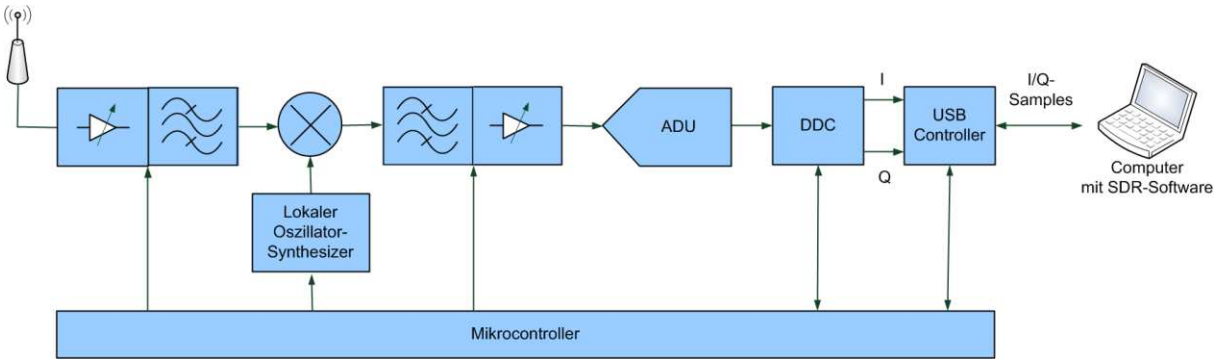


Abbildung 4: Blockdiagramm einer SDR-Empfängervariante (stark vereinfacht)

Das über die Antenne empfangene, analoge Hochfrequenzsignal wird im Tuner der SDR-Hardware mit einem rauscharmen Verstärker verstärkt. Ein Bandpassfilter begrenzt die Signalbandbreite. Das analoge, hochfrequente Bandpass-Signal wird danach mit Hilfe einer Mischstufe auf eine niedrigere Zwischenfrequenz umgesetzt und erneut verstärkt, so dass die nachfolgende Analog-Digital-Umsetzung (ADU) mit einer geringeren Abtastrate erfolgen kann.[4] Trotzdem ist die Abtastrate des ADU deutlich höher festgelegt, als sie für das interessierende Signal benötigt würde. Dies ermöglicht es dem SDR, Eingangssignale verschiedener Funksysteme mit unterschiedlichen Signalbandbreiten zu verarbeiten. Ein Digital Down Converter (DDC) mischt das nunmehr digitale Bandpass-Signal vom Zwischenfrequenzbereich in das Basisband mit der Mittenfrequenz 0 Hz. Nach dieser Frequenzumsetzung liegt ein digitales, komplexes Basisbandsignal mit einer In-Phase- und einer Quadrature-Komponente vor. Anschließend reduziert der DDC durch einen Dezimationsfilter die Datenrate des digitalen, komplexen Basisbandsignals, so dass seine

[4] Für die Analog-Digital-Umsetzung gilt das Nyquist-Shannon-Abtasttheorem, wonach die Abtastrate mehr als das Doppelte der maximalen Signalfrequenz betragen sollte.

Weiterverarbeitung in der SDR-Software mit einem geringeren Rechenaufwand erfolgen kann. Der digitale, komplexe Basisbanddatenstrom wird in Form von I/Q-Samples über die USB-Schnittstelle an den Computer übergeben. Mit der USB-Verbindung kann die SDR-Software den Mikrocontroller der SDR-Hardware ansteuern, um die Empfangsfrequenz, die Signalverstärkung und die gewünschte I/Q-Sample-Rate einzustellen. [5]

3.6.2. SDR-Empfänger-Software

Für den Empfang der Bitcoin Blockchain über Satellit stehen nach der Installation der Blockstream-Software zwei Varianten der SDR-Empfänger-Software zur Verfügung, die Konsolen-Version **rx.py** sowie die GUI-Version **rx_gui.py**. Die Empfänger-Software ist in Python geschrieben. Einige Signalverarbeitungsblöcke, die schneller ausgeführt werden müssen, wurden in C++ implementiert und mit Hilfe von SWIG in die Python-Skripte importiert (siehe 4.1 Software-Installation).

Obwohl Blockstream keine Angaben über die Sendeseite des Satellitensystems macht, lässt ein Blick in das Blockdiagramm der Empfänger-Software **rx_gui.py** ein paar Rückschlüsse auf die eingesetzten Technologien der Bitübertragung zu (Abbildung 5).

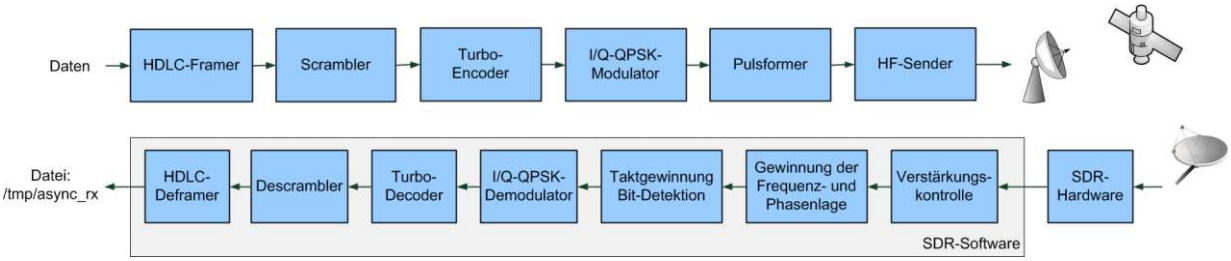


Abbildung 5: Bitübertragung von Blockstream Satellite (vereinfacht)

Für die Bitübertragung werden die Daten im Sender in HDLC-Frames gepackt[6]. In den HDLC-Frames sitzen jeweils am Anfang und am Ende Synchronwörter, mit deren Hilfe sich der Empfänger auf den ankommenden Datenstrom synchronisieren kann. Jeder HDLC-Frame verfügt außerdem über eine CRC-16-Bit-Prüfsumme (Cyclic Redundancy Check), die der Empfänger auswertet, um infolge der Übertragung aufgetretene Bitfehler zu erkennen. Der nachfolgende Scrambler verwürfelt die Bits der HDLC-Frames mit einer definierten Pseudozufallsfolge. Durch die Verwürfelung werden lange Null- oder Eins-Ketten vermieden, wodurch die Bit-Detektion im Empfänger erleichtert wird. Außerdem sorgt die Verwürfelung für eine kontinuierliche Modulation des hochfrequenten Trägersignals, so dass keine Trägerleistungsspitzen durch ein unmoduliertes Trägersignal auftreten und somit andere Funksysteme nicht gestört werden.

---

[5] Moderne Kommunikationssysteme, wie z. B. Wi-Fi, UMTS, LTE, DAB+ und DVB-C/S/T, setzen sogenannte I/Q-Modulatoren ein. Das Prinzip der I/Q-Modulation ist, die binären Daten auf einen In-Phase- und einen Quadrature-Kanal aufzuteilen, deren Daten jeweils auf zwei voneinander unabhängige (orthogonale) Trägerschwingungen (z. B. eine Sinus- und Kosinus-Schwingung) zu modulieren und schließlich die beiden modulierten Trägersignale als Summensignal zu übertragen. Durch ihre Orthogonalität können die Daten beider Kanäle nach der I/Q-Demodulation im Empfänger wiedergewonnen werden.

[6] High-Level Data Link Control (HDLC) ist ein Netzwerkprotokoll der Sicherungsschicht.

Da die Satellitenverbindung keinen Rückkanal besitzt, kann der Empfänger Daten, die er als fehlerhaft erkannt hat, nicht einfach beim Sender erneut anfordern. Damit möglichst wenige Bitfehler beim Empfang auftreten, erhalten die Nutzdaten einen zusätzlichen Fehlerschutz. Es wird eine sogenannte Vorwärtsfehlerkorrektur (engl. Forward Error Correction, FEC) eingesetzt, die den Empfänger bis zu einem gewissen Grad in die Lage versetzt, fehlerhafte Bits nicht nur zu erkennen, sondern sie auch selbst zu korrigieren. Für den Fehlerschutz der Blockstream-Daten setzten die Entwickler eine Turbo-Codierung ein, die z. B. auch in den Mobilfunksystemen der 4. Generation (Long Term Evolution, LTE) verwendet wird. Die Stärke des Fehlerschutzes kann anhand der FEC-Rate abgeschätzt werden. Die FEC-Rate ist das Verhältnis der Datenmenge der Nutzdaten (Datenwortlänge) zur Gesamtdatenmenge inklusive Fehlerschutzdaten (Codewortlänge). Je kleiner die FEC-Rate ist, desto höher ist der Fehlerschutz einzuschätzen. Die Turbo-Codierung erzeugt eine Code-Rate von ca. einem Drittel (Datenwortlänge: 6.144 Bit, Codewortlänge: 18.444 Bit). Nach dem Fehlerschutz wird die Impulsform der Datenbits durch einen Pulsformer verändert. Der Pulsformer bewirkt, dass die höheren Frequenzanteile der Impulse unterdrückt werden und sich somit die Übertragungsbandbreite auf das Notwendigste reduziert. Das Signal wird anschließend einem I/Q-QPSK-Modulator zugeführt. Bei der robusten QPSK (Quadrature Phase Shift Keying) werden jeweils zwei Daten-Bits zu einem Symbol zusammengefasst. Ein Symbol entspricht einer von vier möglichen Phasenlagen des analogen Trägersignals. Das modulierte Trägersignal wird schließlich in die Hochfrequenz umgesetzt und über die Sendeantenne zum Satelliten gesendet.

Im Satellitenempfänger erfolgt zunächst die oben beschriebene Umsetzung des hochfrequenten Satellitenempfangssignals durch Antenne, PLL-LNB und SDR-Empfänger-Hardware in das digitale Basisband. Die weitere Signalverarbeitung übernimmt die SDR-Empfänger-Software. Zunächst reguliert die Empfänger-Software die Signalstärke des Empfangssignals durch eine automatische Verstärkungskontrolle. Nachdem die exakte Frequenz- und Phasenlage des modulierten Trägersignals durch Regelschleifen festgestellt wurde, erfolgt die Rückgewinnung des Signaltaktes und schließlich die Detektion der einzelnen Bits des empfangenen Datenstromes. Im Anschluss werden die zu den Signalverarbeitungsblöcken des Senders jeweils inversen Funktionen in umgekehrter Reihenfolge ausführt. Am Ende der Verarbeitungskette sitzt der HDLC-Deframer, welcher die CRC-Prüfung der empfangenen HDLC-Frames durchführt. Nur diejenigen Nutzdaten, die den CRC-Check passieren, werden über eine Schnittstelle in die Datei /**tmp**/**async_rx** geschrieben, die als Pipe den Eingangsdatenstrom für die Bitcoin-Client-Software **bitcoind** liefert (siehe 4.3 Aktualisierung der Bitcoin-Blockchain über Satellit).

Die theoretisch maximale Bitrate der Daten, welche in die Datei /tmp/async_rx geschrieben werden, lässt sich mit der Gleichung (5) abschätzen: Die SDR-Hardware übergibt der SDR-Software einen komplexen Datenstrom mit einer Symbolrate von 156.250 I/Q-Symbolen pro Sekunde. Der QPSK-Demodulator ordnet jedem Symbol jeweils zwei Bit zu. Durch die Vorwärtsfehlerkorrektur wird die Bitrate entsprechend der FEC-Rate des Turbo-Encoder reduziert (Datenwortlänge: 6.144 Bit, Codewortlänge: 18.444 Bit).

$$R_{/tmp/async_rx} \approx 156250\ symbols/s \cdot 2\ Bit/symbol \cdot \frac{6144}{18444} \approx 104\ kbit/s \tag{5}$$

4. Inbetriebnahme und Test

4.1. Software-Installation

Bei der Auswahl des Computers sollte auf eine ausreichende Prozessorleistung geachtet werden. Wir verwendeten ein Notebook mit 16 GByte RAM und einem Intel® Core™ i5-6300U Prozessor, der mit 2,4 GHz getaktet wird. Möchte man die Bitcoin-Blockchain direkt auf die Festplatte des Notebooks speichern, sollte entsprechend freier Speicherplatz vorhanden sein. Alternativ kann man natürlich auch eine portable Festplatte verwenden, wie wir es getan haben.

Als Betriebssystem nutzten wir Linux Fedora 27 [4]. Während der Installation des Betriebssystems wählt man ein Passwort, das bei einem späteren Update abgefragt wird.

```
# Update des Betriebssystems Linux Fedora
$ sudo dnf update
$ sudo dnf upgrade
```

Der Blockstream-SDR-Empfänger basiert auf dem frei verfügbaren Open-Source-Toolkit GNU Radio [5]. Wir verwendeten die GNU-Radio-Version 3.7.11. Neben der Grundinstallation von GNU Radio werden für den Einsatz der SDR-Empfänger-Hardware der RTL-SDR-Serie die ebenfalls frei verfügbaren GNU-Radio-Blöcke des Pakets GrOsmoSDR benötigt [6]. Beide Pakete können mit dem Fedora-Paketmanager Dnf einfach installiert werden.

```
# GNU Radio installieren
$ sudo dnf install gnuradio gnuradio-devel gr-osmosdr
```

Ist die Installation von GNU Radio und des GrOsmoSDR-Pakets abgeschlossen, kann man dazu übergehen, die für den Blockstream-SDR-Empfänger notwendigen GNU-Radio-Blöcke zu erzeugen. Die Signalverarbeitungsblöcke von GNU Radio sind entweder in der Skriptsprache Python oder in C++ programmiert. Da Blockstream für den SDR-Empfänger eigene GNU-Radio-Blöcke in C++ entwickelte, um eine schnellere Ausführung zu erreichen, müssen zunächst ein C/C++ Compiler und die zugehörigen Entwicklungswerkzeuge installiert werden, die wir für die Erzeugung der GNU-Radio-Blöcke notwendig sind. Außerdem benötigt man das Entwicklungswerkzeug SWIG, mit dessen Hilfe man C++ Programme in eine Skriptsprache wie z. B. Python einbetten kann. SWIG erzeugt auf der Basis der Deklarierungen in den C++ Header Files einen Wrapper-Code, der dann innerhalb des Python-Skripts ausgeführt wird [7].

```
# C/C++ Compiler, Entwicklungswerkzeuge und SWIG installieren
$ sudo dnf groupinstall "C Development Tools and Libraries"
$ sudo dnf install cppunit-devel swig
```

Für den SDR-Empfänger entwickelte Blockstream eigene GNU-Radio-Blöcke, die in den Paketen gr_framers und gr_mods kostenfrei auf GitHub zur Verfügung gestellt werden. GitHub ist ein webbasierter Versionsverwaltungsdienst für Open-Source-Software [8].

```
# Blockstream-Satellite-Pakete mit spezifischen GNU Radio Blöcken erzeugen
$ git clone https://github.com/Blockstream/satellite
$ cd satellite/
$ ./install_gr_framers.sh
$ ./install_mods.sh


# Blockstream-Satellite-Pakete in das Verzeichnis des Satellitenempfängers kopieren
$ cp -r /usr/local/lib64/python2.7/site-packages/framers  ~/satellite/grc
$ cp -r /usr/local/lib64/python2.7/site-packages/mods  ~/satellite/grc


# Pfadvariablen PYTHONPATH und LD_LIBRARY_PATH setzen
$ echo "export PYTHONPATH=/usr/local/lib64/python2.7/sitepackages" >> ~/.profile
$ echo "export LD_LIBRARY_PATH=/usr/local/lib64" >> ~/.profile
$ source ~/.profile
```

Wie bereits erwähnt, nutzt der Blockstream-SDR-Empfänger zur Aktualisierung der Bitcoin-Blockchain die Bitcoin-Client-Software **bitcoind**. In bitcoind ist das FIBRE-Protokoll umgesetzt. Fast Internet Bitcoin Relay Engine (FIBRE) beschleunigt die weltweite Verteilung der Bitcoin-Blöcke, in dem u. a. statt des zuverlässigen und verbindungsorientierten Transportprotokolls TCP (Transmission Data Protokoll) das verbindungslose und schnellere Transportprotokoll UDP (User Data Protocol) eingesetzt wird. Die durch den Einsatz von UDP möglich gewordenen Paketverluste werden mit einer zusätzlich eingeführten Vorwärtsfehlerkorrektur (engl. Forward Error Correction, FEC) der Blockdaten weitestgehend vermieden [9]. Da der FEC-Code an die Blockdaten angehangen wird, vergrößert sich allerdings die zu übertragende Brutto-Datenmenge entsprechend der eingestellten FEC-Rate.

Der Quellcode der Bitcoin-Client-Software bitcoind liegt ebenfalls kostenfrei auf GitHub zur Verfügung.

```
# Erzeugen der Bitcoin-Client-Software bitcoind
$ sudo dnf install openssl-devel libevent-devel libdb4-devel libdb4-cxx-devel
$ git clone https://github.com/bitcoinfibre/bitcoinfibre
$ cd bitcoinfibre/
$ ./autogen.sh
$ ./configure
$ make
$ sudo make install
```

4.2. Aktualisierung der Bitcoin-Blockchain über eine Internetverbindung

Wegen der geringen Netto-Datenrate der Blockstream-Satellitenverbindung werden nur die Blockchain-Blöcke über Satellit ausgestrahlt, die in den letzten 24 Stunden erzeugt wurden. Aus diesem Grund sollte die Blockchain zunächst über eine bestehende Internetverbindung auf ca. 100 Prozent aktualisiert werden. Liegt noch keine Bitcoin-Blockchain auf dem Rechner vor, sind alle Blöcke seit dem Genesis Block, d. h. seit dem 3. Januar 2009, zu aktualisieren, was selbst bei einer sehr guten Internetverbindung einige Stunden in Anspruch nimmt.

In der Abbildung 6 ist das Ende der Blockchain-Aktualisierung über die Internetverbindung dargestellt. Blau markiert sind das Datum und die Nummer (height = 500250) des zuletzt übertragenen Blocks sowie der Fortschritt (100 Prozent).

 # Aktualisierung der Bitcoin-Blockchain über eine Internetverbindung
$ bitcoind -printtoconsole

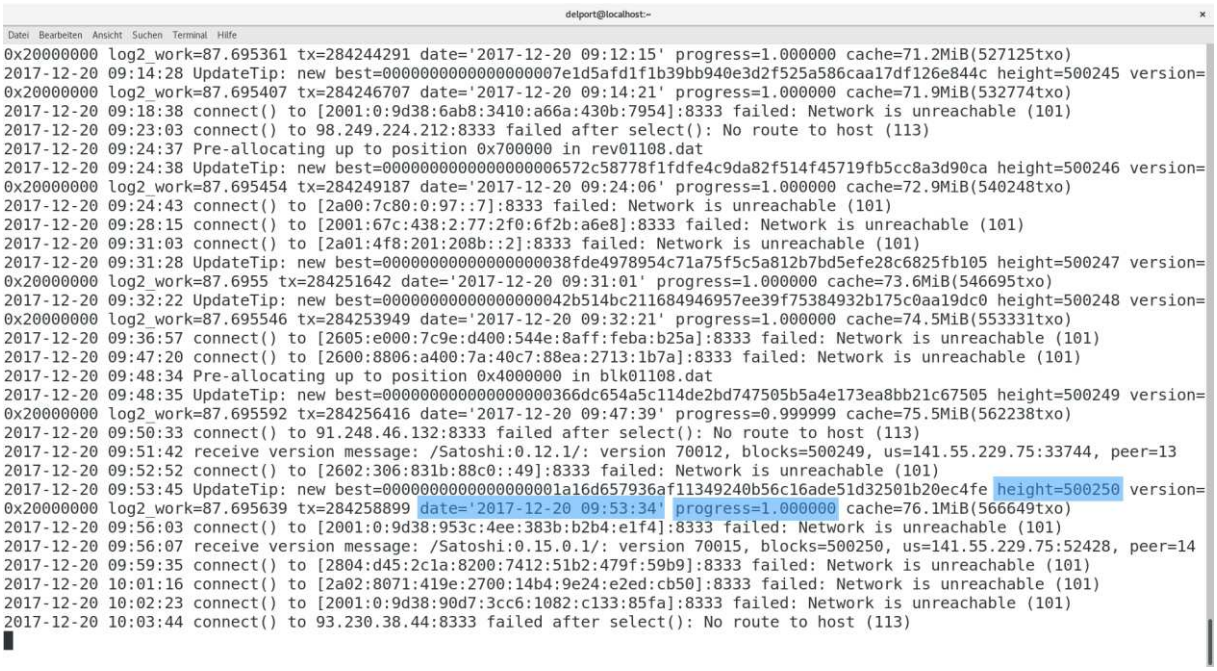


Abbildung 6: Aktualisierung der Bitcoin-Blockchain über eine Internetverbindung

4.3. Aktualisierung der Bitcoin-Blockchain über Satellit

Für den Empfang der Bitcoin-Blockchain über Satellit stehen nach der Software-Installation zwei Varianten der Blockstream-SDR-Empfängersoftware zur Verfügung - **rx.py** sowie **rx_gui.py**.

4.3.1. Empfang des Blockstream-Satellitensignals mit rx_gui.py

Die Anwendung **rx_gui.py** besitzt eine grafische Oberfläche. Mit ihr kann man zunächst abschätzen, ob und wie gut das Blockstream-Satellitensignal empfangen werden kann. Beim Start von rx_gui.py müssen die SAT-Zwischenfrequenz in Hz und die gewünschte Verstärkung des Signals in dB als Parameter übergeben werden. Da das Blockstream-Satellitensignal im unteren Frequenzband ausgestrahlt wird und somit die LNB-Oszillatorfrequenz 9.750,00 MHz beträgt, liegt die SAT-Zwischenfrequenz nach Gleichung (4) bei 1.754,02 MHz.

 # Kann das Blockstream-Satellitensignal empfangen werden?
$ python rx_gui.py - - freq 1754020000 - - gain 50

Die folgenden Screenshots von rx_gui.py wurden während eines erfolgreichen Empfangs des Blockstream-Satellitensignals aufgenommen. Die Abbildung 7 zeigt das modulierte Trägersignal im Zeitbereich nach der Feststellung seiner genauen Frequenzlage durch die SDR-Empfänger-Software (siehe 3.6.2 SDR-Empfänger-Software).

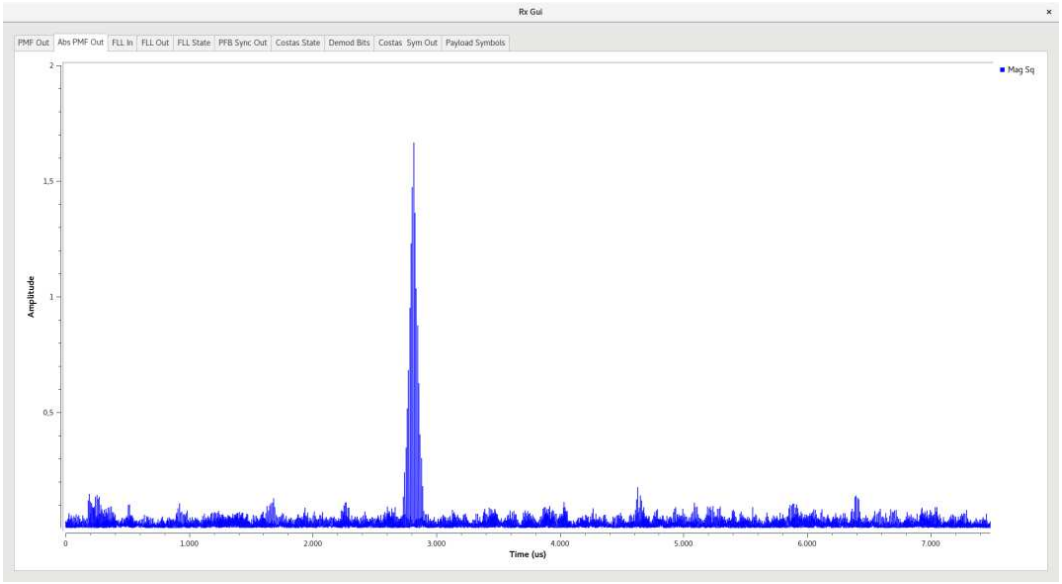


Abbildung 7: Moduliertes Trägersignal nach der Feststellung seiner Frequenzlage

Das Verhalten des Trägersignals ist ein erster, wichtiger Hinweis dafür, ob das Blockstream-Satellitensignal korrekt empfangen wird. Bleibt die Lage des Trägersignals über der Zeit konstant, wurde seine Frequenzlage durch den SDR-Empfänger detektiert. Ist das nicht der Fall, bewegt sich das Trägersignal über der Zeitachse. Eine mögliche Ursache für dieses Verhalten könnte die ungenügende Frequenzstabilität des verwendeten LNB sein.

Die Abbildung 8 stellt die Verteilung der relativen Signalstärke des Blockstream-Basisbandsignals über der Frequenz dar.



Abbildung 8: Relative Signalstärke des Blockstream-Basisbandsignals über der Frequenz

Der mittlere blaue Graph zeigt die momentan gemessene relative Signalstärke. Der obere und der untere Graph stellen die über einen längeren Messzeitraum erreichten Maximalwerte bzw. Minimalwerte dar. Zu erkennen ist, dass die Signalstärke in einem Intervall von ca. 30 dB schwankt, d. h. linear um bis zu einem Faktor von 1000.

Die Abbildung 9 zeigt das Konstellationsdiagramm der QPSK-modulierten Daten. Die große Streuung der QPSK-Symbole über ihren idealen Positionen deutet auf ein störendes Breitbandrauschen hin, das für Übertragungen über eine geostationäre Satellitenverbindung typisch ist. Ein zu starkes Breitbandrauschen verursacht nach der QPSK-Demodulation Bitfehler, die durch den Turbo-Decoder erkannt und bestenfalls korrigiert werden können. Anderenfalls fallen die betreffenden Daten durch die CRC-Prüfung des HDLC-Defamer, der den betreffenden HDLC-Frame verwirft. Um den zugehörigen Block in die Blockchain schreiben zu können, muss der SDR-Empfänger auf die erneute Aussendung des HDLC-Frame warten (siehe 3.6.2 SDR-Empfänger-Software).

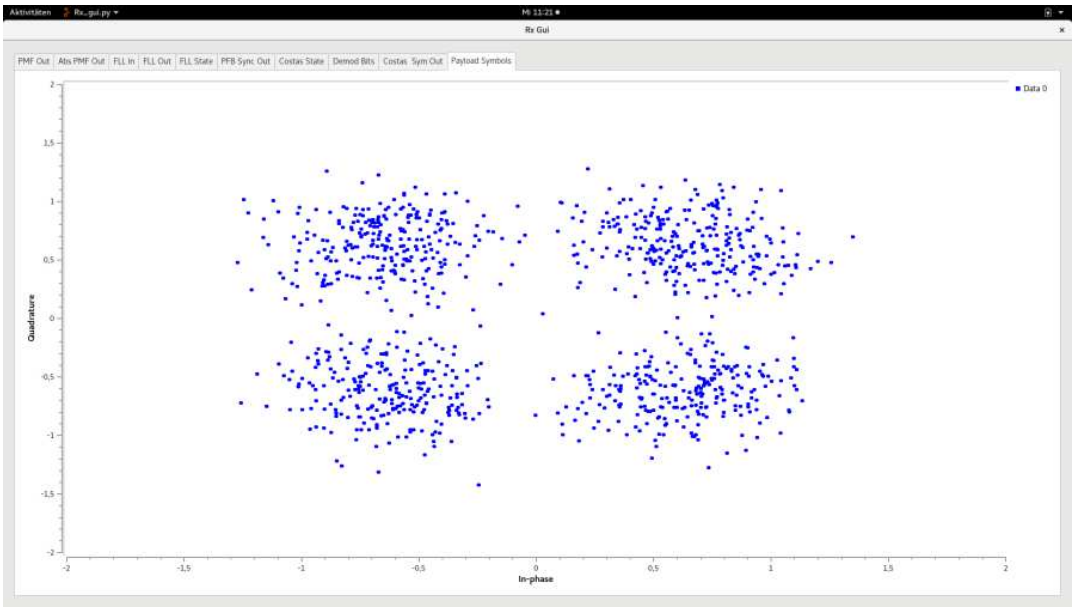


Abbildung 9: QPSK-Konstellationsdiagramm

4.3.2. Aktualisierung der Bitcoin-Blockchain mit rx.py

Für die Aktualisierung der Bitcoin-Blockchain mithilfe des Blockstream-Satellitenempfängers empfiehlt sich die Anwendung **rx.py**, da sie keine GUI besitzt und somit den Empfangsrechner weniger belastet.

 # Blockstream-Satellitensignal empfangen
$ python rx.py - - freq 1754020000 - - gain 50

Die Abbildung 10 zeigt die Konsolenausgabe nach dem Start von rx.py. Schon nach kurzer Zeit synchronisiert sich die SDR-Empfangssoftware auf die übertragenen Daten-Frames, worauf die Konsolenausgabe „Frame synchronization acquired“ hinweist. Die Ausgabe des Nutzdatenstromes in die Datei /tmp/async_rx beginnt.

```
gr-osmosdr v0.1.x (0.1.5git) gnuradio 3.7.11
built-in source types: file fcd rtl rtl_tcp uhd hackrf rfspace redpitaya
Using device #0 Realtek RTL2838UHIDIR
Found Elonics E4000 tuner
Exact sample rate is: 1250000.002070 Hz
File exists
and is pipe, opening ...
Press Enter to quit: [work] Error in distance btw peak 0 and 1:  158(expected 9352)
[work] Avg peak distance:        0.000000
[work] Var peak distance:        0.000000
[work] Error in distance btw peak 1 and 2:       9483(expected 9352)
[work] Avg peak distance:        0.790000
[work] Var peak distance:        123.574928
[work] Error in distance btw peak 2 and 3:       9483(expected 9352)
[work] Avg peak distance:        48.205002
[work] Var peak distance:        445200.343750
[work] Error in distance btw peak 3 and 4:       9483(expected 9352)
[work] Avg peak distance:        95.620003
[work] Var peak distance:        885814.875000
[work] Error in distance btw peak 4 and 5:       9483(expected 9352)
[work] Avg peak distance:        143.035004
[work] Var peak distance:        1321989.625000
[work] Error in distance btw peak 5 and 6:       6330(expected 9352)
[work] Avg peak distance:        190.449997
[work] Var peak distance:        1753747.000000
[work] Error in distance btw peak 6 and 7:       9247(expected 9352)
[work] Avg peak distance:        222.099991
[work] Var peak distance:        1940279.250000
[work] Error in distance btw peak 7 and 8:       9253(expected 9352)
[work] Avg peak distance:        268.334991
[work] Var peak distance:        2343361.250000
[work] Error in distance btw peak 8 and 9:       9263(expected 9352)
[work] Avg peak distance:        314.600006
[work] Var peak distance:        2742836.250000
[work] Error in distance btw peak 9 and 10:      9310(expected 9352)
[work] Avg peak distance:        360.915009
[work] Var peak distance:        3139071.750000
[work] Error in distance btw peak 10 and 11:     9339(expected 9352)
[work] Avg peak distance:        407.464996
[work] Var peak distance:        3535347.500000
##### Frame synchronization acquired #####
```

Abbildung 10: Start der SDR-Empfangssoftware rx.py

Nach dem erfolgreichen Start der Anwendung rx.py ist es zweckmäßig, ihre Prozesspriorität von „Normal“ auf „Sehr hoch“ zusetzen.

Nun kann die Blockchain-Cient-Software **bitcoind** erneut gestartet werden.

 # Aktualisierung der Bitcoin-Blockchain über Blockstream Satellite
$ bitcoind -fecreaddevice=/tmp/async_rx -printtoconsole -connect=0 -fecstat=60

Mit der Option *-fecreaddevice* wird der Pfad der Datei /**tmp**/**async_rx** übergeben, in den die bereits laufende SDR-Empfangssoftware **rx.py** den empfangenen Datenstrom schreibt. Die Option *-connect=0* unterbindet den automatische Verbindungsaufbau zu einem anderen FIBRE-Knoten. Mit der Option *-fecstat=60* kann man sich ca. jede Minute die erzielte mittlere Datenrate der UDP-Verbindung über die Konsole ausgegeben lassen.

Die Abbildung 11 zeigt die Konsolenausgabe nach dem Start von bitcoind. Blau markiert sind die Nummer (height = 500250) und das Datum des letzten, über die Internetverbindung aktualisierten Blocks.

In der Abbildung 12 ist die Konsolenausgabe der sich anschließenden, fortlaufenden Aktualisierung der Blöcke 500251 bis 500258 dargestellt. Neben den Blocknummern sind die Ausgaben der mittleren UDP-Datenraten blau markiert. Bei den zum Zeitpunkt der Messung vorherrschenden Empfangsbedingungen ermöglicht die Satellitenverbindung eine mittlere UDP-Datenrate von ca. 40 kbit/s. Mit dieser Datenrate ist die kontinuierliche Aktualisierung der Bitcoin-Blockchain per Blockstream Satellite möglich.

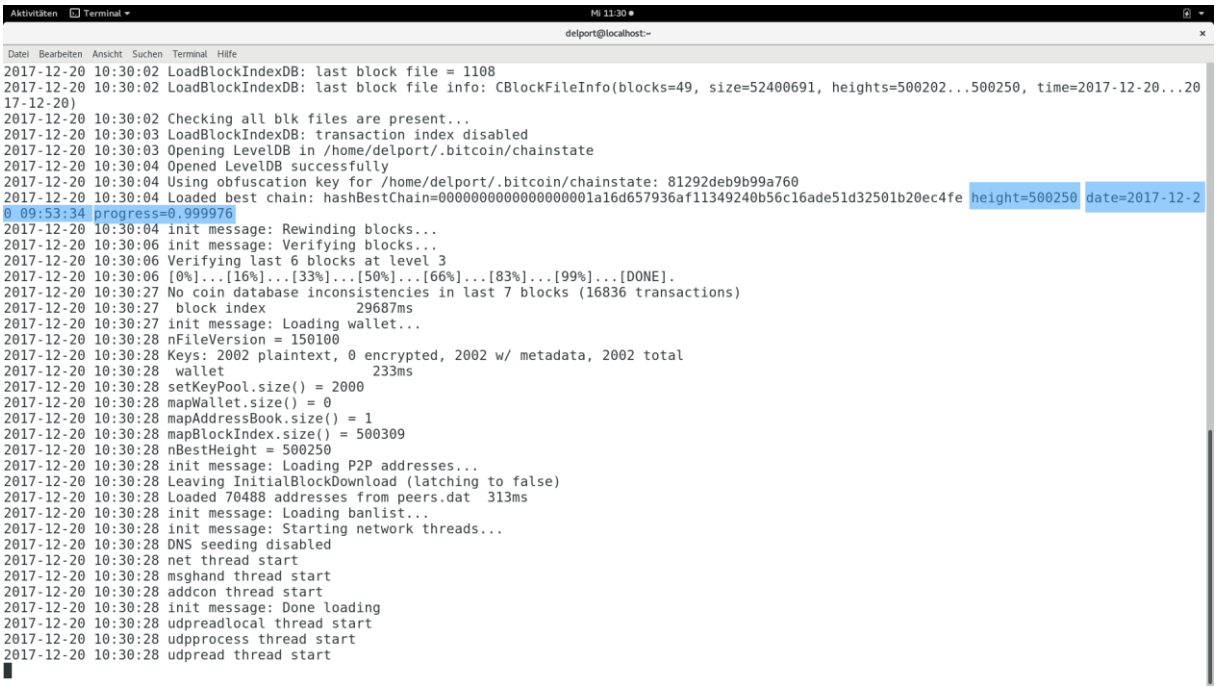


Abbildung 11: Screenshot der Konsole nach dem Start von bitcoind



Abbildung 12: Fortlaufende Aktualisierung der Blöcke 500251 bis 500258

5. Fazit

In einem abschließenden Fazit möchten wir unsere wesentlichen Erfahrungen beim Aufbau und Betrieb der Blockstream-Satellitenempfangsanlage kompakt zusammenfassen.

Das Blockstream-Signal für Europa sendet der Telstar 11N 37,5° W im unteren Frequenzband auf 11.504,02 MHz mit vertikaler Polarisation aus.

Zur Ausrichtung der Satellitenantenne sind für den betreffenden Empfangsort Azimut, Elevation und Skew (LNB-Kippwinkel) zu bestimmen und einzustellen.

Aufgrund der vergleichsweise niedrigen Elevation für den Empfang des Telstar 11N 37,5° W in Deutschland empfehlen wir, einen Satellitenspiegel mit einem Durchmesser von 120 cm einzusetzen.

Bei der Auswahl eines geeigneten LNB für den Empfang des Blockstream-Satellitensignals muss auf die Frequenzstabilität bei der Erzeugung der Lokaloszillatorschwingung im LNB geachtet werden. Wir empfehlen, einen LNB mit einer integrierten Phase-Locked Loop (PLL) zu verwenden.

Da das Blockstream-Signal im unteren Frequenzband mit vertikaler Polarisation ausgestrahlt wird, genügt es, den LNB über eine passive Frequenzweiche (SAT-Combiner) mit einer Gleichspannung von +14 V zu versorgen. Bei der Auswahl der Frequenzweiche ist zu beachten, dass sie nur über einen der beiden Pfade eine Gleichspannung (DC) passieren lässt. An den Pfad ohne den DC-Durchlass schließt man das SDR-Gerät an, damit die SDR-Hardware durch die +14V-Gleichspannung nicht beschädigt wird.

Die SDR-Empfänger-Software von Blockstream Satellite ist kompatibel für SDR-Hardware-Module der RTL-SDR-Serie.

Bei der Auswahl des Empfangsrechners sollte auf ausreichende Prozessorleistung und genügend freie Festplattenkapazität zur Speicherung der vollständigen Bitcoin-Blockchain geachtet werden.

Als Betriebssystem nutzten wir Linux Fedora 27. Der SDR-Empfänger nutzt das frei verfügbare Open-Source-Toolkit GNU Radio. Wir verwendeten die GNU-Radio-Version 3.7.11.

Wegen der geringen Netto-Datenrate der Blockstream-Satellitenverbindung werden nur die Blockchain-Blöcke über Satellit ausgestrahlt, die in den letzten 24 Stunden erzeugt wurden. Deshalb sollte die Blockchain zunächst über eine bestehende Internetverbindung auf ca. 100 Prozent aktualisiert werden.

Für die Aktualisierung der Blockchain setzten wir die Bitcoin-Client-Software **bitcoind** ein.

Für den Empfang der Bitcoin-Blockchain über Satellit stehen nach der Software-Installation zwei Varianten der Blockstream-SDR-Empfängersoftware zur Verfügung - **rx.py** sowie **rx_gui.py**.

Die Anwendung **rx_gui.py** besitzt eine grafische Oberfläche. Mit ihr kann man feststellen, ob und wie gut das Blockstream-Satellitensignal empfangen werden kann.

Zur Ausführung von **rx_gui.py** bzw. **rx.py** müssen die SAT-Zwischenfrequenz in Hz und die gewünschte Verstärkung des Eingangssignals in dB der SDR-Empfänger-Hardware als Parameter übergeben werden. Da das Blockstream-Satellitensignal im unteren Frequenzband ausgestrahlt wird und somit die LNB-Oszillatorfrequenz 9.750,00 MHz beträgt, liegt die SAT-Zwischenfrequenz bei 1.754,02 MHz.

 # Kann das Blockstream-Satellitensignal empfangen werden?
$ python rx_gui.py - - freq 1754020000 - - gain 50

Für die Aktualisierung der Bitcoin-Blockchain mithilfe des Blockstream-Satellitenempfängers empfiehlt sich die Anwendung **rx.py**, da sie keine GUI besitzt und somit den Empfangsrechner weniger belastet.

 # Blockstream-Satellitensignal empfangen
$ python rx.py - - freq 1754020000 - - gain 50

Nach dem erfolgreichen Start der Anwendung rx.py ist es zweckmäßig, ihre Prozesspriorität von „Normal“ auf „Sehr hoch“ zusetzen.

Den über die Satellitenverbindung empfangenen Datenstrom gibt die Anwendung **rx.py** in die Datei /**tmp**/**async_rx** aus, die als Pipe den Eingangsdatenstrom für die Bitcoin-Client-Software **bitcoind** liefert.

 # Aktualisierung der Bitcoin-Blockchain über Blockstream Satellite
$ bitcoind -fecreaddevice=/tmp/async_rx -printtoconsole -connect=0 -fecstat=60

Obwohl wir für die Satellitenempfangsanlage einen vergleichsweisen großen Satellitenspiegel von 120 cm Durchmesser einsetzen, war bei schlechten Empfangsbedingungen das Verhältnis der Nutzsignalleistung zur Stör- bzw. Rauschsignalleistung zu gering, so das im Empfänger trotz der eingesetzten Vorwärtsfehlerkorrektur (FEC-Rate ≈ 0,34) zu viele Bitfehler entstanden. Da jedoch keine fehlerhaften Daten in die Blockchain gelangen dürfen, werden die Daten vor ihrem Aussenden in Frames gepackt und jeder Frame mit einer CRC-16-Bit-Prüfsumme versehen. Bevor der Satellitenempfänger rx.py die Daten über die Datei /tmp/async_rx an die Bitcoin-Client-Software bitcoind übergibt, wird diese CRC-Summe geprüft. Passiert während des Empfangs eines Blockes nur ein Daten-Frame den CRC-Check nicht, kann der Block nicht in die Blockchain geschrieben werden, d. h. der Empfänger muss auf das erneute Aussenden des Blockes bzw. des Daten-Frame warten. Eine kontinuierliche Aktualisierung der Bitcoin-Blockchain ist unter diesen Umständen nicht möglich.

An Tagen mit sehr guten Empfangsbedingungen konnten wir dagegen UDP-Datenraten von 40 bis 50 kbit/s über die Satellitenverbindung erzielen. Mit diesen Datenraten ist eine kontinuierliche Aktualisierung der Bitcoin-Blockchain über Blockstream Satellite gegeben.

6. Referenzen

1. **Back, Adam.** *https://blockstream.com/2017/08/15/announcing-blockstream-satellite.html.*

2. **Blockstream-Satellite-FAQ.** *https://blockstream.com/satellite/faq/.*

3. **E.Blossom.** *http://www.linuxjournal.com/article/7319.*

4. **Fedora.** *https://getfedora.org.*

5. **GNURadio.** *https://www.gnuradio.org.*

6. **Osmocom.** *http://osmocom.org/projects/sdr/wiki/GrOsmoSDR.*

7. **SWIG.** *http://www.swig.org.*

8. **GitHub.** *https://github.com.*

9. **FIBRE.** *http://bitcoinfibre.org.*

10. **Bitcoin-Blockchain-Size.** *https://blockchain.info/de/charts/blocks-size.*