

Erfahrungen mit einer aktuellen Fernsteuerung

Inhaltsverzeichnis

| | |
|--|----|
| 1. Vorwort..... | 3 |
| 2. Wie es dazu kam..... | 3 |
| 3 Das Funkmodul RFM63W..... | 6 |
| 3.1 Die technischen Daten..... | 6 |
| 3.2 Das Blockschaltbild..... | 7 |
| 3.3 Die möglichen Übertragungsverfahren..... | 8 |
| 3.3.1 Direkte Übertragung..... | 8 |
| 3.3.2 Gepufferte Übertragung..... | 8 |
| 3.3.3 Paket-Modus..... | 8 |
| 3.4 Die Kommunikation mit dem Fernsteuermodul..... | 9 |
| 4. Das Eigenbauprojekt..... | 12 |
| 4.1 Die Fernbedienung..... | 12 |
| 4.1.1 Kodierung der Tasten..... | 12 |
| 4.1.2 Hilfsmittel zur Fehlersuche..... | 14 |
| 4.2 Der Fernsteuerempfänger..... | 14 |
| 4.2.1 Das Problem mit den wenigen noch freien Anschlüssen..... | 14 |
| 4.2.2 Das Problem mit dem Verhalten des FIFOs..... | 14 |
| 5. Böse Überraschungen..... | 16 |
| 5.1 Sporadisch kommen falsche Daten an..... | 16 |
| 5.2 Der Empfänger bleibt taub..... | 18 |
| 5.3 Ein- und Ausgaben sind um einen Tastendruck versetzt..... | 18 |
| 6. Eine überraschende Zahl..... | 19 |
| 7. Zusammenfassung..... | 20 |
| 8. Quellenverzeichnis..... | 20 |

1. Vorwort

Vor längerer Zeit hatte ich mich mit einem Aufbau beschäftigt für den eine Fernsteuerung gebraucht wurde. Dafür hatte ich zwei kostengünstige Module (Sender und Empfänger) für das 868 MHz Band besorgt. Die Beschreibung für diese Module passte auf ein DIN A5-Blatt. Entsprechend einfach und intuitiv war der Betrieb dieser Module gewesen. Für ein aktuelles Projekt brauchte ich wieder eine Fernsteuerung und hatte mir dafür Fernsteuermodule (< 2€ pro Stück) besorgt. Beim ersten Blick in das Datenblatt zeigte sich dann, dass die neuen Module um Größenordnungen komplexer waren. Von den Erfahrungen soll hier berichtet werden.

2. Wie es dazu kam

Im Jahr 2014 war ich von einem Bekannten worden gefragt, ob ich ein Gerät bauen könne, mit dessen Hilfe eine nicht verriegelte Terrassentür von außen verriegeln und wieder freigeben könne. Dazu sollte auf der Innenseite ein Balken mit Hilfe dieses Gerätes quergelegt oder angehoben werden. Das sollte mit einer Schnur gemacht werden, die auf einer Spule auf- und abgewickelt werden sollte. Der Antrieb dazu sollte dann per Fernsteuerung von außen bedient werden können. Wegen einer langen Reihe von Zusatzwünschen entwickelte sich das Ganze dann zu einem wahrhaften Mammutprojekt. Hier möchte aber nur auf die Fernsteuerung selbst eingehen.

Für die Fernsteuerung hatte ich mir bei einem Elektronikhändler für 18€ einen Platinensatz besorgt, bestehend aus einem 868MHz Sender und einem dazu passenden Empfänger.

Der Sender hatte nur einen Digitaleingang über den man ein Signal mit einer Maximalfrequenz von etwa 10 kHz einspeisen konnte. Der Empfänger (ein Pendelaudion) hatte einen Digitalausgang an dem man das übertragene Signal abnehmen konnte.

Problem: Bei ausgeschaltetem Sender oder zu großer Entfernung wurde am Empfängerausgang ein Rauschsignal ausgegeben.

Lösung: Auf der Sendeseite musste ein Signal eingespielt werden das sich stark von weißem Rauschen unterscheidet. Auf der Empfangsseite musste ein Dekoder installiert werden, der erkennt ob wirkliche Information übertragen wird oder nicht. Das ging dann natürlich nicht ohne Mikrocontroller auf beiden Seiten. Damit ließen sich dann natürlich auch eine Menge anderer Funktionen einbauen wie z.B. erzeugen und auswerten einer Prüfsumme.

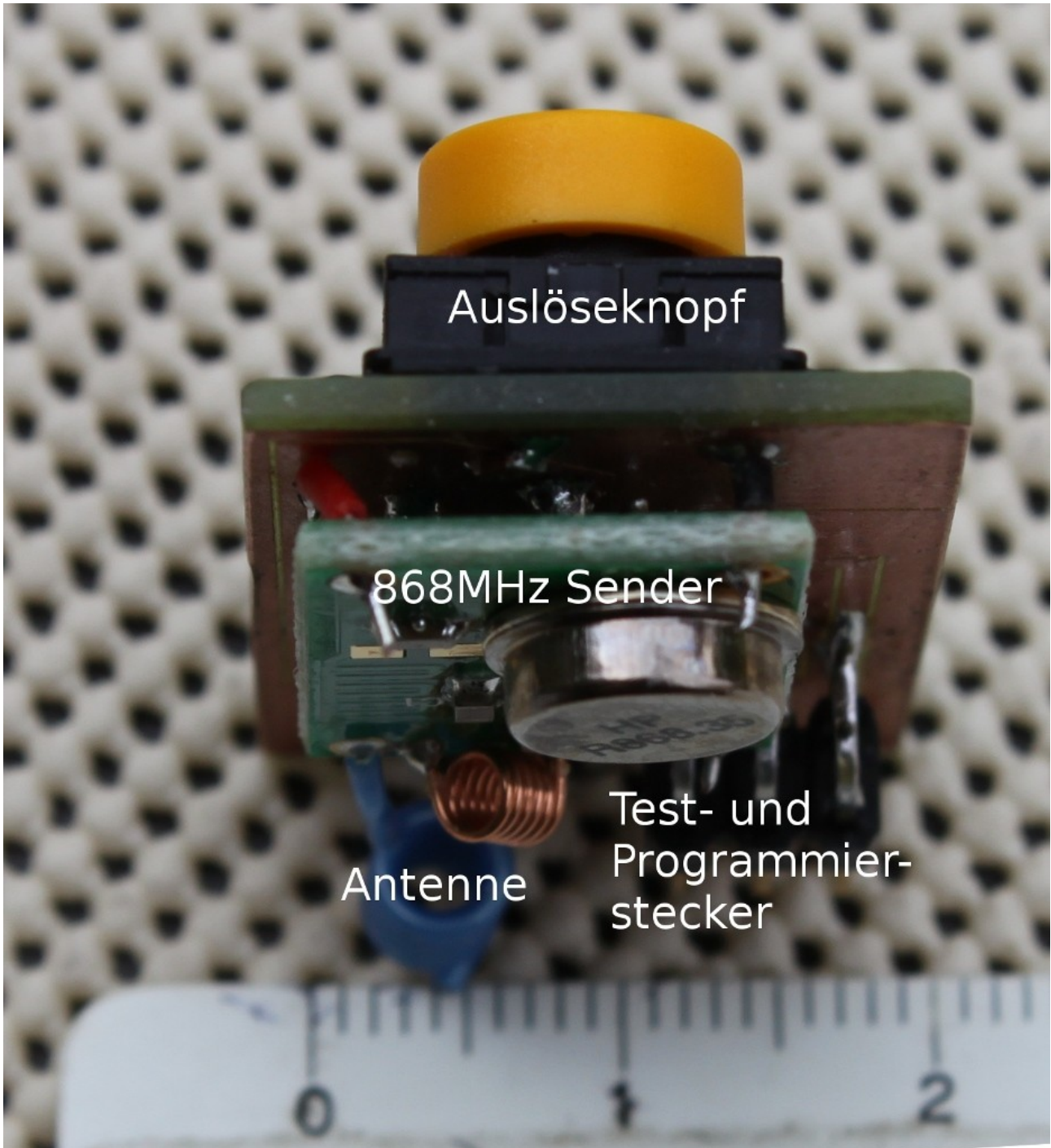


Bild 1: Der Fernsteuersender auf einer Platine mit dem Mikrocontroller, dem Bedientaster und der Batterie

Erfahrungen mit einer aktuellen Fernsteuerung

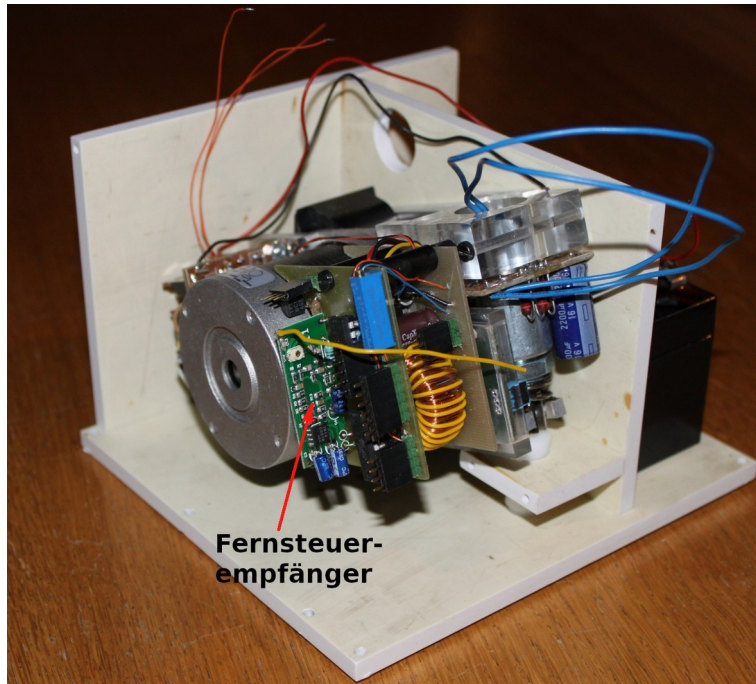


Bild 2: Empfänger der Fernsteuerung des Gerätes

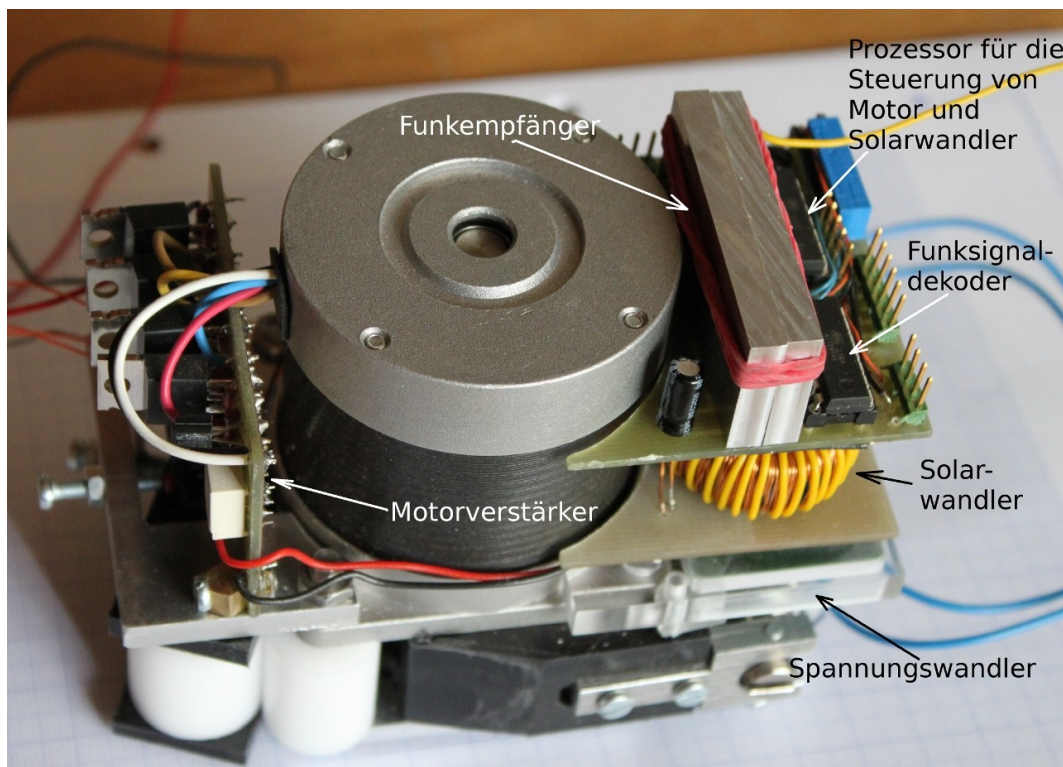


Bild 3: Geräteansicht mit Funkempfänger (im Gehäuse) und dem Mikrocontroller

Dieses Jahr (2023) wurde ich von einem Bekannten gefragt ob ich etwas bauen könne, das auch

Erfahrungen mit einer aktuellen Fernsteuerung

wieder eine Fernsteuerung erfordert. In Erinnerung an die Fernsteuerung aus dem Jahr 2014 dachte ich, dass ich auf die damaligen Erfahrungen zurückgreifen könne und vielleicht sogar Teile der damals geschriebenen Mikrocontrollerprogramme übernehmen könne. Eine Suche im Internet brachte eine größere Zahl solcher Fernsteuermodule zum Vorschein. Ich habe mir einige Module mit der Typbezeichnung RFM63W besorgt mit der Meinung dass man bei Modulen die weniger als 2€ kosten wohl nichts falsch machen könne und dass diese kostengünstigen etwa wie die bekannten aus dem Jahr 2014 zu betreiben seien. Ein Blick in das zugehörige 90-seitige Daten-“Blatt“ zeigte aber, dass die Welt inzwischen ganz anders aussieht.

Nachdem man diese aktuellen Fernsteuermodule aber für weit mehr benutzen kann als nur die Tastennummer einer Fernbedienung zu übertragen sollten hier die Erfahrungen mit diesem Modul beschrieben werden. Ob diese Erfahrung sich auch mit denen anderer Typen decken, habe ich noch nicht untersucht.

3 Das Funkmodul RFM63W

Hardwaremäßig ist das Funkmodul ein kleines Kärtchen (16 mm x 20 mm). Jeweils 8 Anschlüssen auf zwei Seiten.

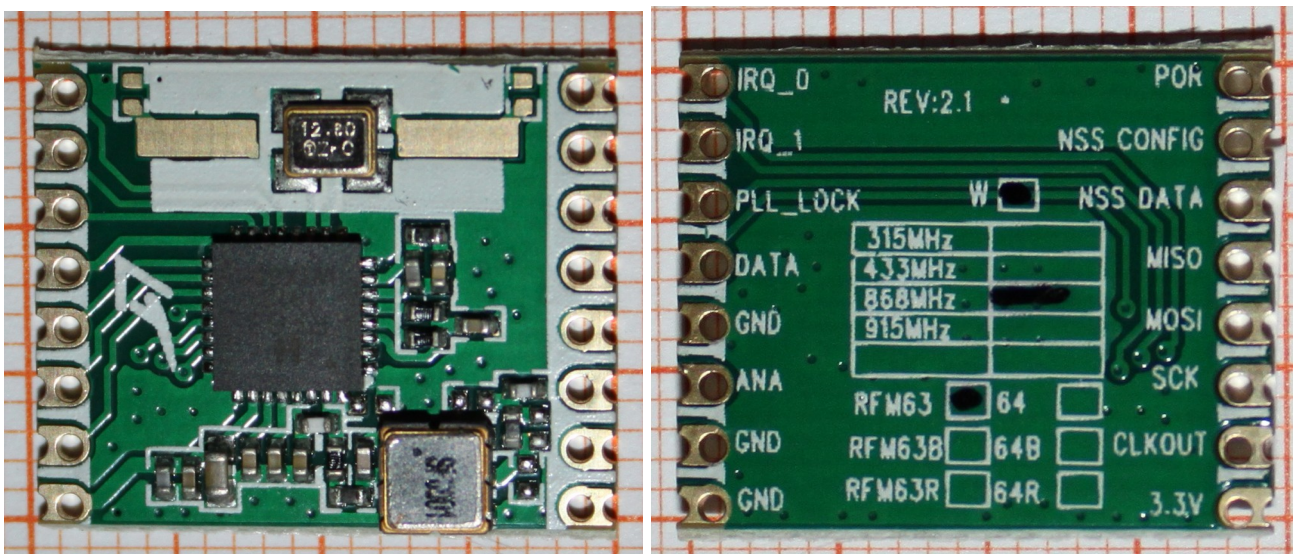


Bild 4: Vorder- und Rückseite der neuen Fernsteuermodule

Ursprünglich wollte ich die Module in meinem Aufbau steckbar machen. Dazu hätte ich Abstände von 2,54 mm gebraucht, die Anschlüsse haben aber nur 2 mm Abstand.

3.1 Die technischen Daten

Die Versorgungsspannung kann zwischen 2,1 V und 3,6 V liegen.

Die maximale Übertragungsrate liegt bei 200 kb/s

Die Sendeleistung ist maximal 12dBm

Die Empfindlichkeit -111 dBm bei 25kb/s oder -105 dBm bei 200 kb/s

3.2 Das Blockschaltbild

Das vereinfachte Blockschaltbild zeigt, dass das Kärtchen als Sender und als Empfänger betrieben werden kann. Der Empfänger ist kein Plendelaudion (wie bei der Fernsteuerung aus dem Jahr 2014) sondern ein vollständiger Doppelsuperhet-RX.

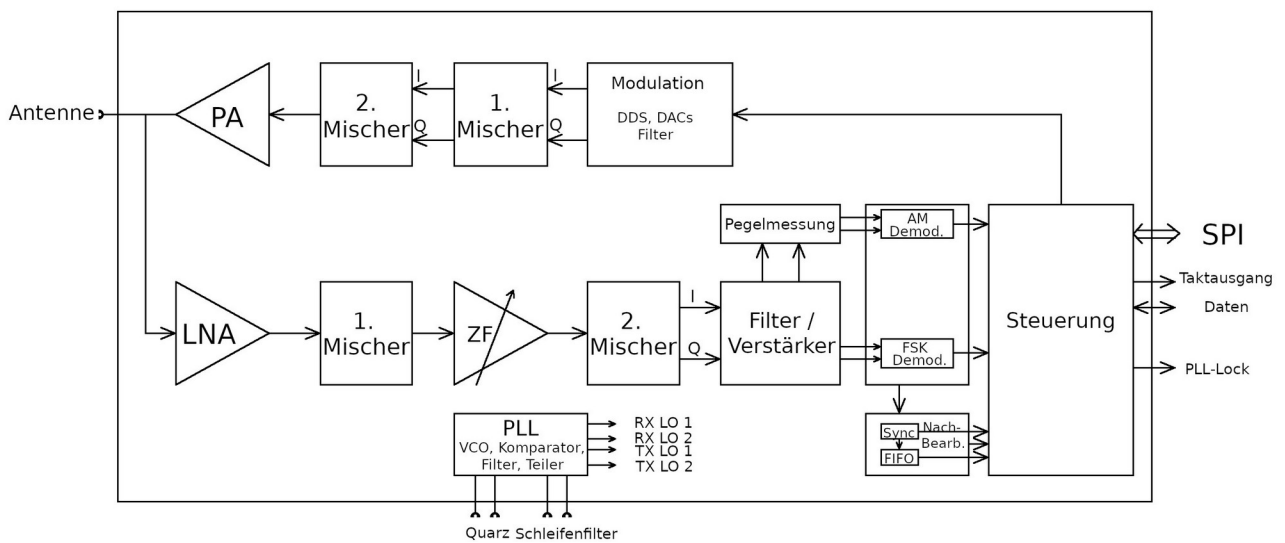


Bild 5: Blockschaltbild Funkfernsteuermodul

Dieses Bild ließ schon vermuten, dass es mit dem Einschalten der Stromversorgung alleine nicht getan ist. Ein erster Versuch bestätigte diese Vermutung, außer dass das Kärtchen bei 3,3 V etwa 0,3uA verbraucht hat, hat es sich ansonsten wie ein toter Fisch verhalten.

3.3 Die möglichen Übertragungsverfahren

Sieht man sich dazu noch die Beschreibung an, zeigt sich, dass das Fernsteuermodul drei Modi der Datenübertragung ermöglicht:

3.3.1 Direkte Übertragung

Das auf der Sendeseite auf der „Daten“-Leitung eingegebenen Signal taucht auf der Empfangsseite ohne Verzögerung auf der „Daten“-Leitung auf.

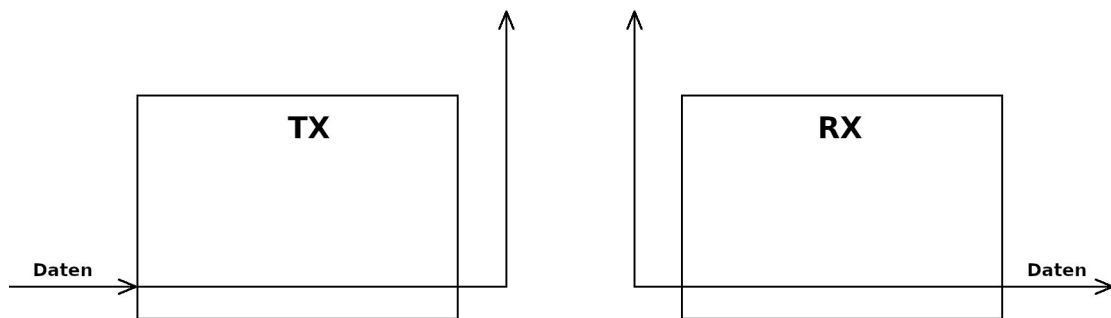


Bild 6: Direkte Datenübertragung

3.3.2 Gepufferte Übertragung

Die auf der Sendeseite über die serielle Schnittstelle (SPI) eingegebenen Daten werden auf der Empfängerseite in einem FIFO-Speicher abgelegt und können dort zeitversetzt über die SPI ausgelesen werden.

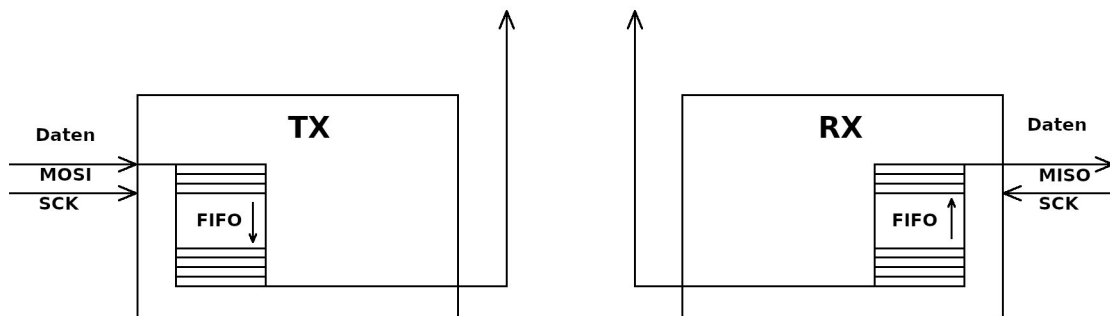


Bild 7: Gepufferte Übertragung

3.3.3 Paket-Modus

Die auf der Sendeseite über die serielle Schnittstelle (SPI) eingegebenen Daten werden zu Datenpaketen mit fixer oder variabler Länge zusammengefasst. Jedes der Pakete kann noch mit Zusatzinformation wie einen Vorspann zur Synchronisation und Auswahl eines von mehreren Empfängern, einer Datenadresse und einer CRC-Prüfsumme versehen werden. Diese Datenpakete werden auf der Empfängerseite ausgewertet und gültige Daten im FIFO abgelegt. Außerdem kann

Erfahrungen mit einer aktuellen Fernsteuerung

die Übertragung der Nutzdaten mittelwertfrei gemacht werden, entweder mit einer NRZ-Kodierung oder mit einer speziellen Faltung mit dem Generatorpolynom $x^9 + x^5 + 1$.

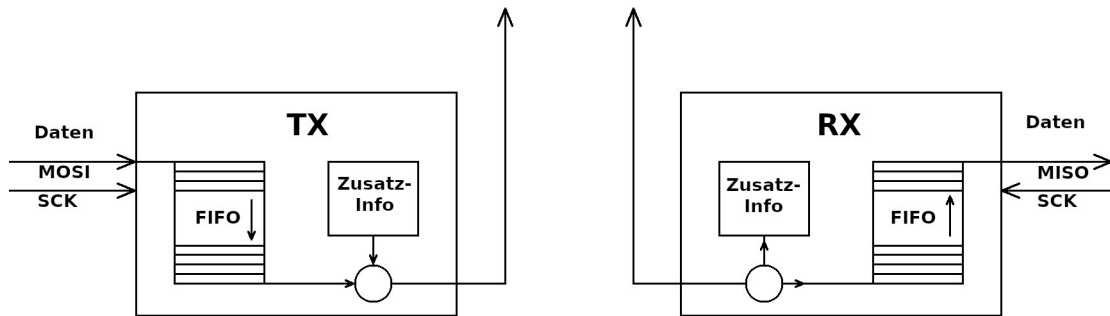


Bild 8 Paketübertragung



Bild 9: Paketaufbau

Nachdem der Paket-Modus gegenüber den anderen am meisten Komfort bietet, hatte ich mich entschlossen genau diesen Modus zu benutzen.

3.4 Die Kommunikation mit dem Fernsteuermodul

Um die Fernsteuermodule so einstellen zu können dass damit ein funktionierender Betrieb möglich wird, ist eine große Zahl von Parametern zu setzen (Gesamtanzahl 76) wobei eine Reihe davon schon mit Standardwerten vorbesetzt ist. Diese Parameter bleiben aber nicht erhalten sondern müssen bei jeder Spannungswiederkehr neu gesetzt werden. So etwas lässt sich nur mit einem Mikrocontroller realisieren. Die für den Betrieb nötigen Verbindungen zwischen Fernsteuermodul und Mikrocontroller sehen damit so aus:

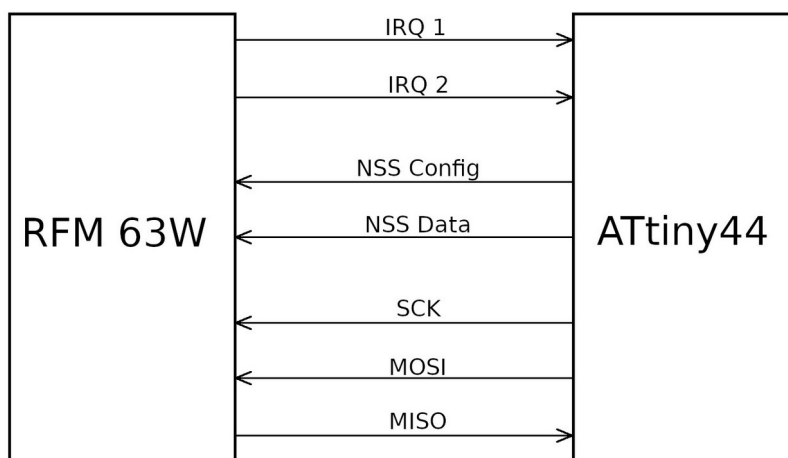


Bild 10: Verbindungen zwischen Fernsteuermodul und Mikrocontroller

Erfahrungen mit einer aktuellen Fernsteuerung

Die Leitungen haben dabei folgende Funktionen

- NSS Data Gibt an, ob Nutzdaten (Sendung oder Empfang) übertragen werden sollen
- NSS Config Gibt an, ob Konfigurationsdaten übertragen werden sollen
- IRQ 1 Leitung, die (konfigurierbare) Zustände der Fernsteuerung anzeigt (z.B. FIFO voll, FIFO leer, CRC-Fehler...).
- IRQ 2 Wie IRQ 1
- SCK Takt für die zu übertragenden Daten
- MISO Daten Richtung Fernsteuerung → Mikrocontroller
- MOSI Daten Richtung Mikrocontroller → Fernsteuerung

Im Datenblatt werden für die Signale auf diesen Leitung genaue Vorgaben gemacht mit dem Hinweis, dass die Vorgaben genau einzuhalten sind. Da es ohne genaue und vermutlich recht aufwendige Untersuchung nicht möglich war festzustellen ob die standardmäßige Schnittstelle des Mikrocontrollers in allen denkbaren Fällen zu der der Fernsteuerung passt, habe ich beschlossen, diese Schnittstelle selbst auszuprogrammieren.

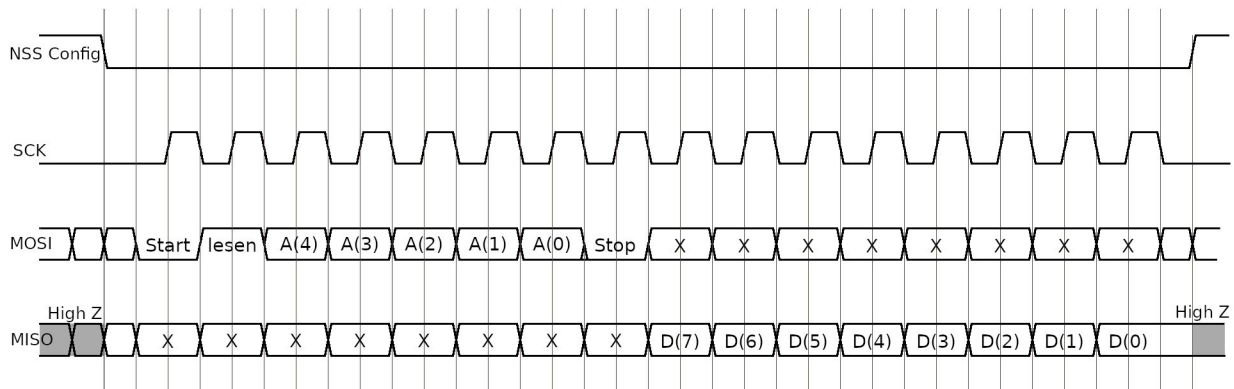


Bild 11: Zeitdiagramm für das Lesen der Konfigurationsdaten

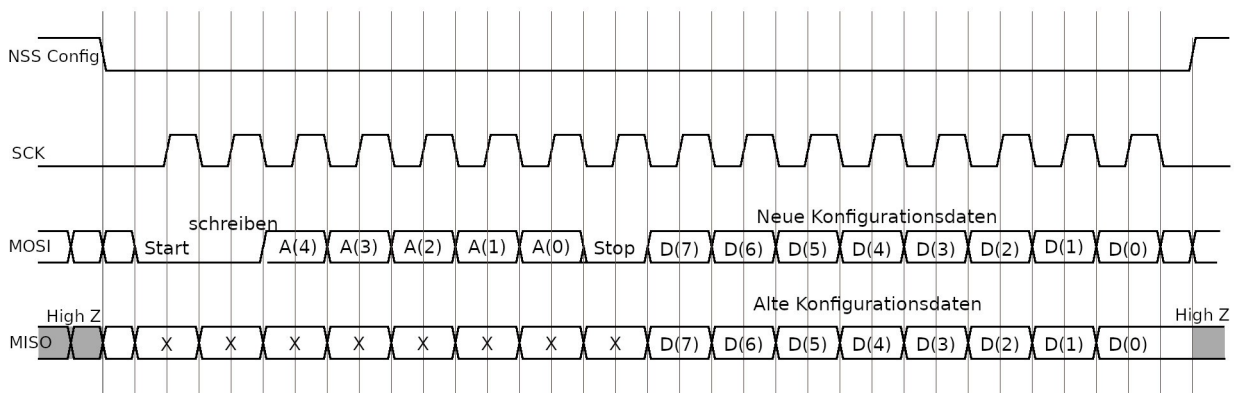


Bild 12: Zeitdiagramm für das Schreiben der Konfigurationsdaten

Erfahrungen mit einer aktuellen Fernsteuerung

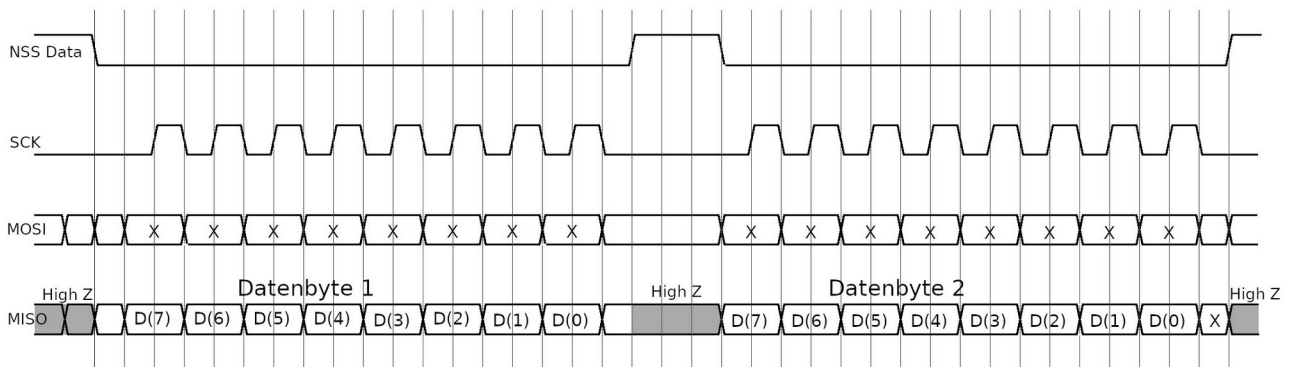


Bild 13: Zeitdiagramm für das Lesen der Nutzdaten

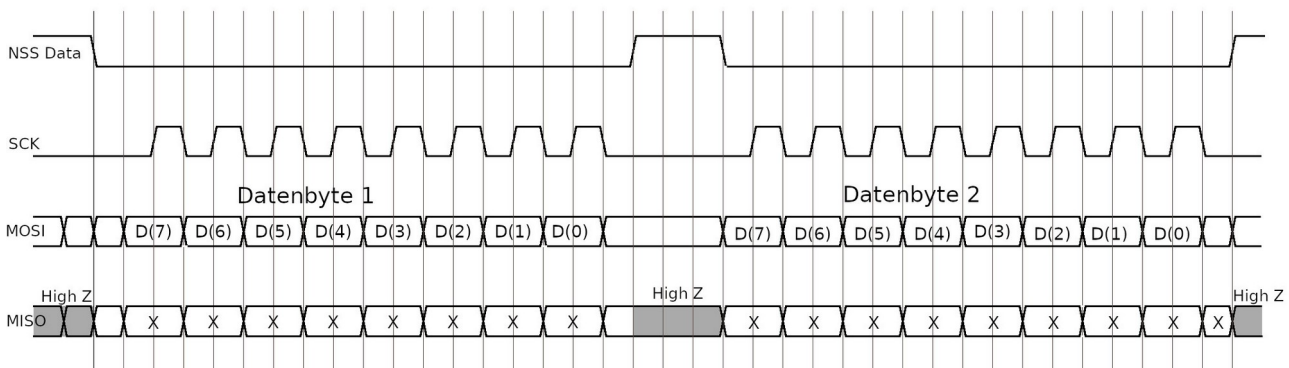


Bild 14: Zeitdiagramm für das Schreiben der Nutzdaten

Nach jeder Spannungswiederkehr müssen die Parameter neu gesetzt werden. Dabei werden aber nur die geschrieben, die von den Standardwerten abweichen

Parameter für den Empfänger

| Adresse | Wert | Bedeutung |
|---------|------|---|
| 00 | 10 | Schlaf-Modus |
| 01 | 8C | FSK, Paket Modus, ZF 0dB |
| 06 | 7D | PLL R1 Wert für 868 MHz |
| 07 | 64 | PLL P1 Wert für 868 MHz |
| 08 | 16 | PLL S1 Wert für 868 MHz |
| 0D | 80 | IRQ 0 „FIFO nicht leer“, IRQ1 CRC OK |
| 16 | A5 | 1. Byte Syncwort |
| 17 | C3 | 2. Byte Syncwort |
| 1C | 10 | Manchester Kodierung AUS, Paketlänge 16 Bytes |
| 00 | 70 | Empfangsmodus |

Erfahrungen mit einer aktuellen Fernsteuerung

Alle anderen Parameter bleiben die unveränderten Defaultwerte

Parameter für den Sender

| Adresse | Wert | Bedeutung |
|---------|------|---|
| 01 | 8C | FSK, Paket Modus, ZF 0dB |
| 06 | 7D | PLL R1 Wert für 868 MHz |
| 07 | 64 | PLL P1 Wert für 868 MHz |
| 08 | 16 | PLL S1 Wert für 868 MHz |
| 16 | A5 | 1. Byte Syncwort |
| 17 | C3 | 2. Byte Syncwort |
| 1C | 10 | Manchester Kodierung AUS, Paketlänge 16 Bytes |
| 00 | 10 | Schlaf-Modus |

Alle anderen Parameter bleiben die unveränderten Defaultwerte

4. Das Eigenbauprojekt

Mit diesen Modulen und dem Paketmodus sollte das Projekt realisiert werden. Die Sendeseite sollte so wie eine übliche Fernbedienung aussehen und mit einer 3V Knopfzelle versorgt werden können. Der Empfänger mit der zugehörigen Applikation sollte ebenfalls in einem kleinen Gehäuse untergebracht werden, die Versorgung wahlweise mit einer 9V Blockbatterie oder extern 10V ... 24V.

4.1 Die Fernbedienung

Die Fernbedienung (also die Senderseite) hat 14 Tasten. Mit jedem Tastendruck sollte ein Telegramm geschickt werden, das die Tastennummer enthält. Um den Stromverbrauch möglichst gering zu halten, sollte der Sender mit dem Tastendruck eingeschaltet und nach der Übertragung des Datenpakets wieder abgeschaltet werden.

4.1.1 Kodierung der Tasten

Als Mikrocontroller hatte ich den ATtiny44 ausgesucht weil ich mit diesem Typ schon Erfahrungen z.B. mit der Platine für die Lange Nacht der Wissenschaften hatte. Dieser Chip hat 14 Anschlüsse:

- zwei für Stromversorgung (Vcc und Masse)
- einen für den RESET. Diese Leitung ich auch für die Programmierung des Chips erforderlich und kann deshalb nicht für andere Zwecke benutzt werden
- sieben Anschlüsse (siehe Bild 6) sind für die Ankopplung der Fernsteuerung zu reservieren (NSS Data, NSS Config, SCK, MISO, MOSI, IRQ 1 und IRQ 2)
- einer ist notwendig um für die Fehlersuche „ein Fenster in den Chip“ zu haben.

Es bleiben also noch 3 freie Anschlüsse übrig.

Damit lassen sich binär nur 8 Tasten programmieren. Wenn man es genau nimmt nur 7 denn es muss ja auch der Zustand „keine Taste gedrückt“ erkannt werden.

Erfahrungen mit einer aktuellen Fernsteuerung

Die Lösung

Anstatt nur 0 und 1 – also 0V und 3,3V zu unterscheiden wird eine Widerstandskette mit 10 Widerständen aufgebaut und die Spannungen 0,33V, 0,66V... 3,3V mit den Tastern abgegriffen und auf einen der freien Eingänge gelegt. Mit zwei solcher Widerstandsketten lassen sich dann 20 Tasten kodieren.

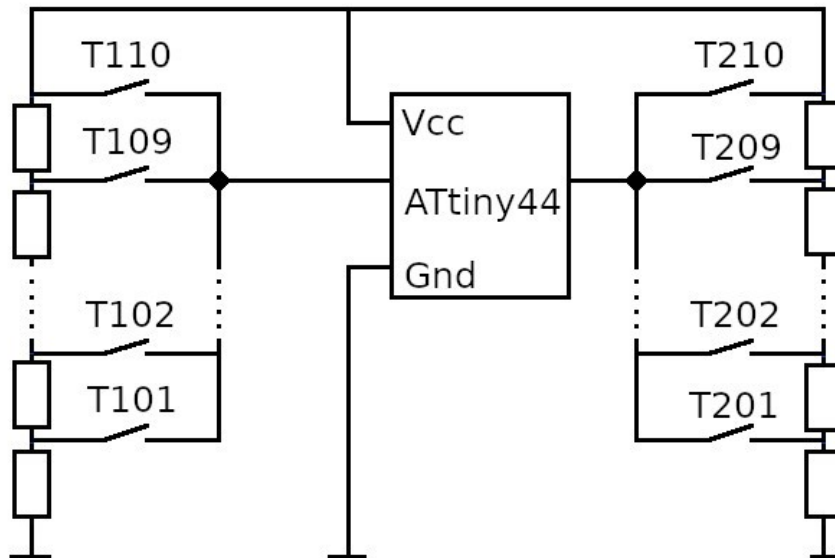


Bild 15 Kodierung der Tasten in der Fernbedienung

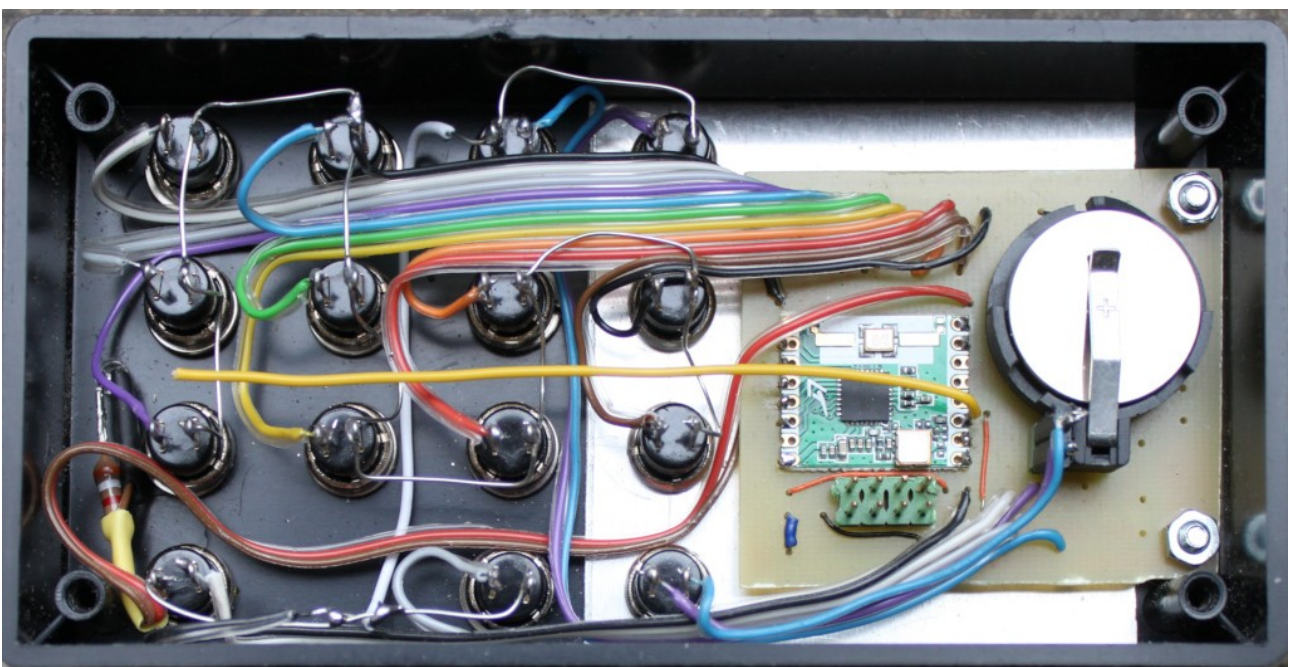


Bild 16: Blick in die Fernbedienung

4.1.2 Hilfsmittel zur Fehlersuche

Auch wenn man sich noch so sehr bemüht – beim Programmieren schleichen sich unvermeidlich Fehler ein. Außerdem blieben trotz sehr ausführlicher Beschreibung des Fernsteuermoduls noch Fragen offen die sich nur durch gezielte Tests beantworten ließen. Um auch über die eine der beiden freien Leitungen möglichst viel Information zu bekommen habe ich noch ein kleines Unterprogramm eingebaut, das den Inhalt eines im Programm übergebenen Bytes als Hexazeichen in CW-Codierung ausgibt. Damit ließ sich mit einem kleinen Piezo-Piepser abhören was im Programm an einer vordefinierten Stelle abläuft.

4.2 Der Fernsteuerempfänger

Der Fernsteuerempfänger ist beim Einschalten zu parametrieren, danach sind die empfangenen Tastenpakete auszulesen und weiter zu verarbeiten.

4.2.1 Das Problem mit den wenigen noch freien Anschlüssen

Genau wie auf der Sendeseite sind am Mikrocontroller nur noch drei Anschlüsse frei. Damit lassen sich die eigentlich gewünschten Reaktionen auf die Tastenbedienungen nicht realisieren. Die freien Leitungen müssen deshalb zur Übertragung der Tasteninfos zu einem weiteren Mikrocontroller genutzt werden.

4.2.2 Das Problem mit dem Verhalten des FIFOs

Auf der Empfangsseite werden die Daten in einen FIFO geschrieben, also einen Speicher mit einer festen Länge. Die Minimale Länge des FIFOs liegt bei 16 Byte. Es müssen also 16 Byte gesendet werden bis die erste Tasteninfo durch den Speicher durchgelaufen ist und am Speicherausgang erscheint. Würde man mit einem Tastendruck jeweils nur 1 Byte senden, würde der Empfänger immer um 16 Tastenbedienungen hinterherhinken. Die Lösung ist, auf der Sendeseite jeden Tastendruck 16 mal zu senden und auf der Empfangsseite auf die „Speicher-voll“-Meldung hin, den FIFO 16 mal auszulesen.

Der Mikrocontroller mit der eigentlichen Applikation

Die Applikation auf dem dritten Mikrocontroller soll hier nicht weiter erläutert werden. Dieser dritte Mikrocontroller erhält über die auch selbst ausprogrammierte serielle Schnittstelle die Tastentelegramme und führt die damit verbundenen Aktionen aus. Im unten gezeigten Bild sieht man ein breites Flachbandkabel von dem aber nur wenige der verfügbaren Leitungen (Takt, Datenleitung, Masse und Stromversorgung) benutzt werden.

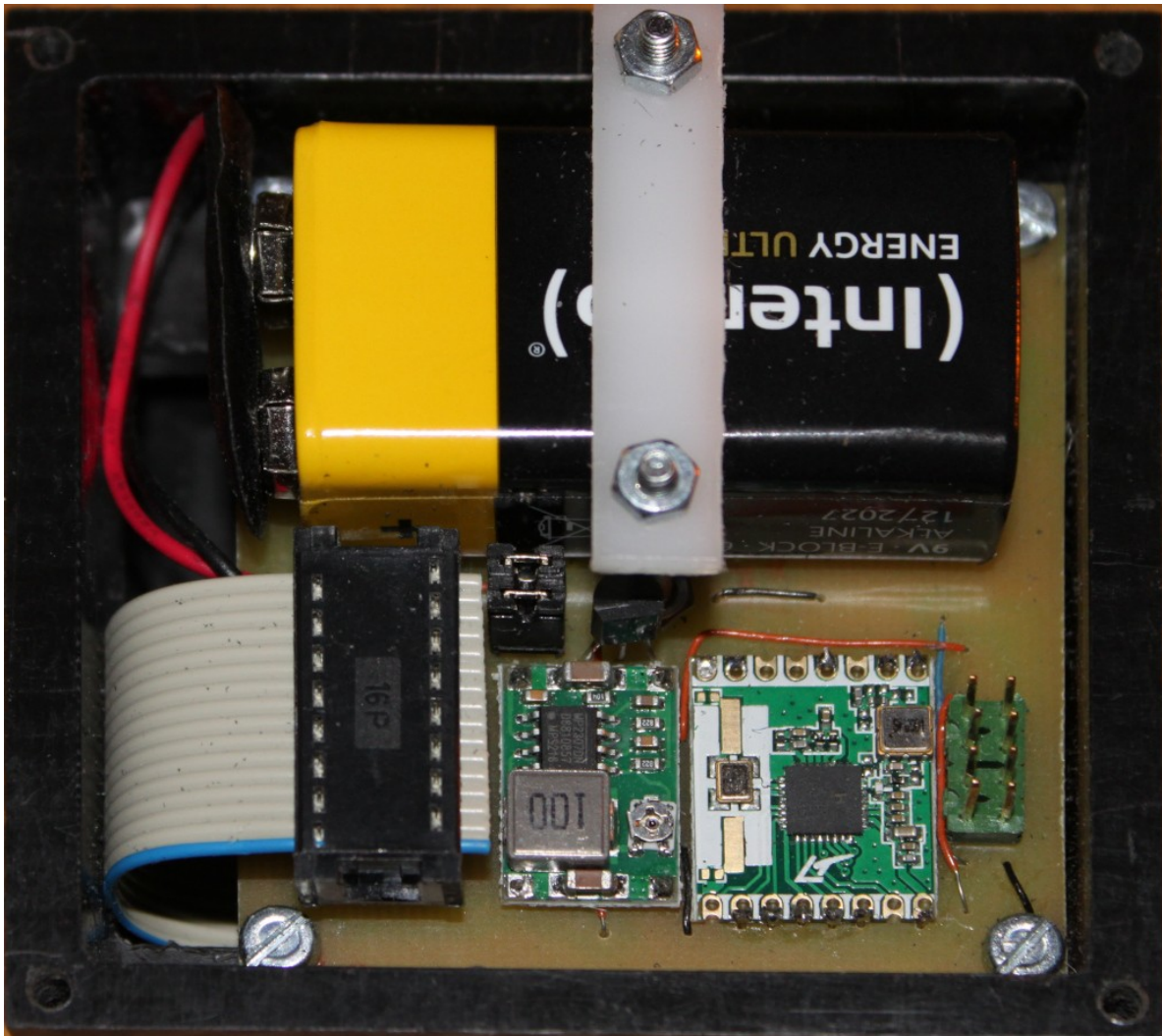


Bild 17: Blick in den Fernsteuerempfänger RX-Seite

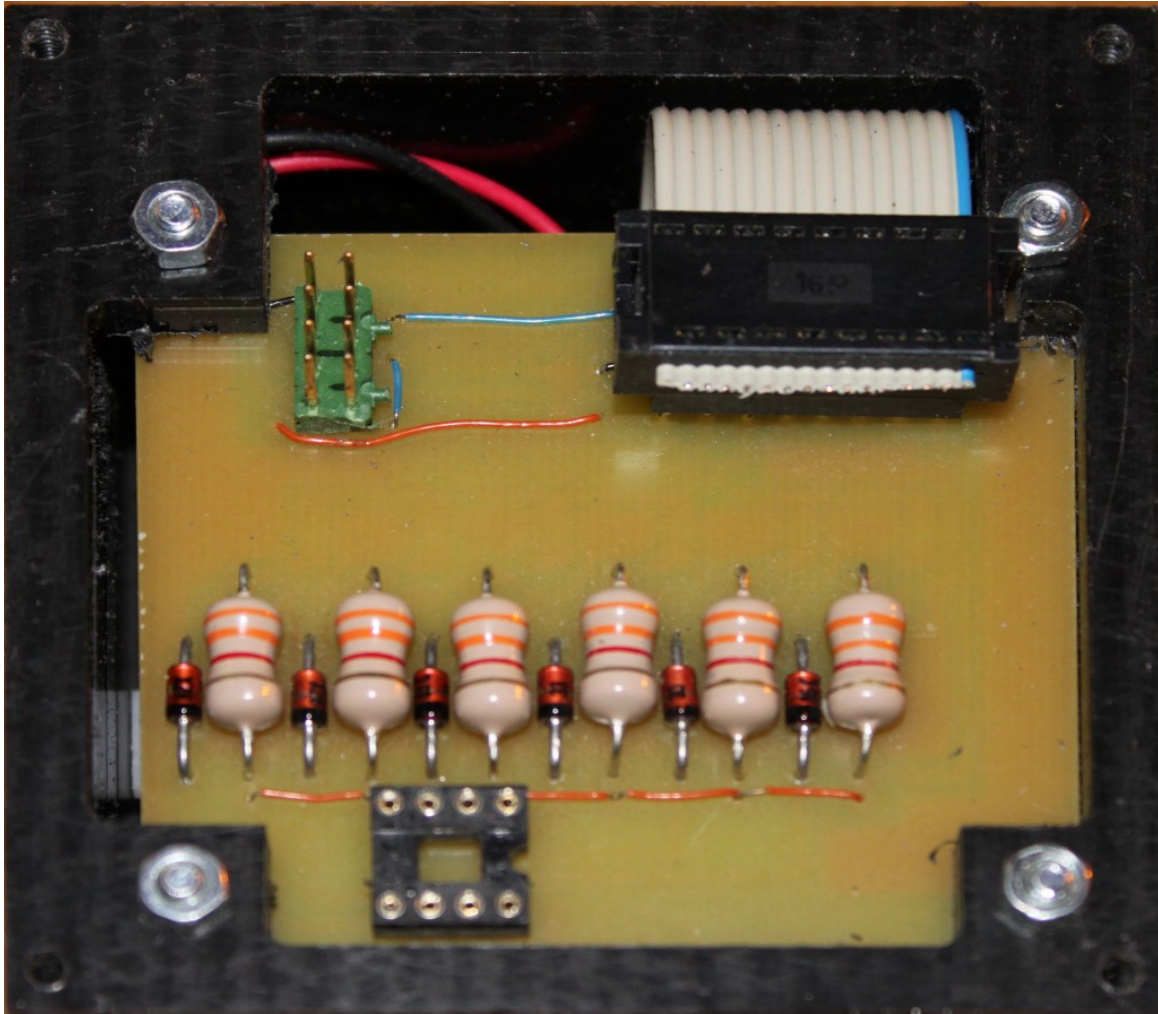


Bild 18 Blick in den Fernsteuerempfänger Applikationsseite

5. Böse Überraschungen

Auch bei diesem „Projekt“ hat sich das gezeigt, was allen größeren technischen Vorhaben eigen zu sein scheint: Es tauchen Probleme auf deren Behebung viel Zeit braucht, oft auch weil die Fehlersuche in die falsche Richtung geht. Und das war bei der Fernsteuerung nicht anders.

5.1 Sporadisch kommen falsche Daten an

Um Strom zu sparen, sollten die Modi des Fernstellersenders (wie im Datenblatt beschrieben) vom Status IDLE über STANDBY zu SENDEN und zurück zu STANDBY umgeschaltet werden. Für den Zeitablauf gibt es im Datenblatt genaue Vorgaben.

Erfahrungen mit einer aktuellen Fernsteuerung

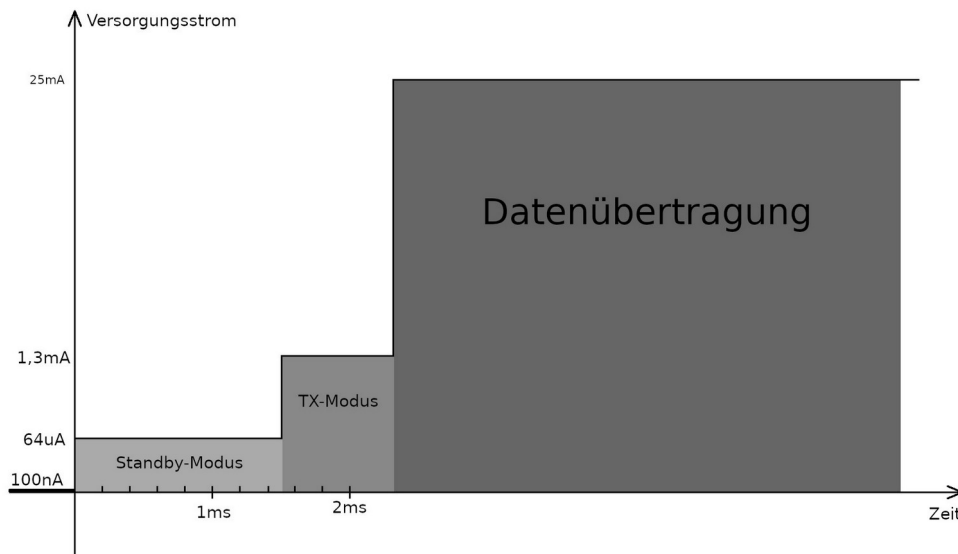


Bild 19: Zeitlicher Übergang vom Schlafmodus in den Sendebetrieb

Die einzige Unsicherheit dabei war die Zeit, die für die Datenübertragung wirklich gebraucht wird um den Zeitpunkt zu treffen zu dem wieder in STANDBY zurückgeschaltet werden darf. Die Idee war gewesen, einen der IRQ-Ausgänge so zu parametrieren, dass er den Zustand „FIFO leer“ anzeigt und damit auch dass der Sendevorgang abgeschlossen ist. Irgendwie wollte das aber nicht funktionieren, nach längerer Suche hat sich herausgestellt, dass der IRQ-Ausgang aus unerfindlichen Gründen nicht das macht was durch die Parameter vorgegeben wurde. Der nächste Ansatz war, zu schätzen wie lange eine Übertragung wohl dauern würde, auf diese Zeit noch ein paar Millisekunden draufzulegen und danach in den STANDBY-Zustand zurückzuschalten.

Bei einer Übertragung von einzelnen (isolierten) 2-Byte langen Telegrammen funktionierte das. Immer wenn mehr als 2 Byte in einer schnellen Folge übertragen werden sollten, tauchten auf der Empfangsseite ab dem Byte 2 fehlerhafte Daten auf.

Verändern des Zeitablaufs brachte keine Besserung, auch die Vermutung, dass das Sendesignal in die Datenleitungen des Mikrocontrollers einkoppeln konnte ich nicht verifizieren. Nach langer vergeblicher Suche stellte ich dann auch die eigene Mikrocontroller-Software in Frage und wurde prompt fündig: Die Zellen die für die Versorgung der seriellen Schnittstelle ausgelesen wurden, wurde durch die neuen Daten überschrieben bevor die aktuellen Daten ausgelesen waren. Nach der Fehlerbehebung kamen alle Daten auf der Empfangsseite korrekt an. Für die Sendezeiten wurden aber weiterhin fixe Zeiten benutzt, weil das Problem mit über das IRQ-Signal gesteuerten Zeiten nicht behoben werden konnte.

5.2 Der Empfänger bleibt taub

Die ersten Versuche mit den Fernsteuermodulen wurde mit fliegenden Aufbauten gemacht. Im zweiten Schritt wurde dann kleine Platinen gebaut auf denen auf der einen Seite das Fernsteuermodul mit Drähtchen aufgelötet wurde. Auf dieser Seite waren auch die bedrahteten Bauelemente und die Steckerleisten montiert worden. Auf der Rückseite wurden der Mikrocontroller und alle SMD-Bauelemente aufgelötet. Für diese Aufbauten wurde neue Fernsteuermodule verwendet um das Verhalten in Fehlerfällen mit dem der „fliegenden“

Testaufbauten vergleichen zu können.

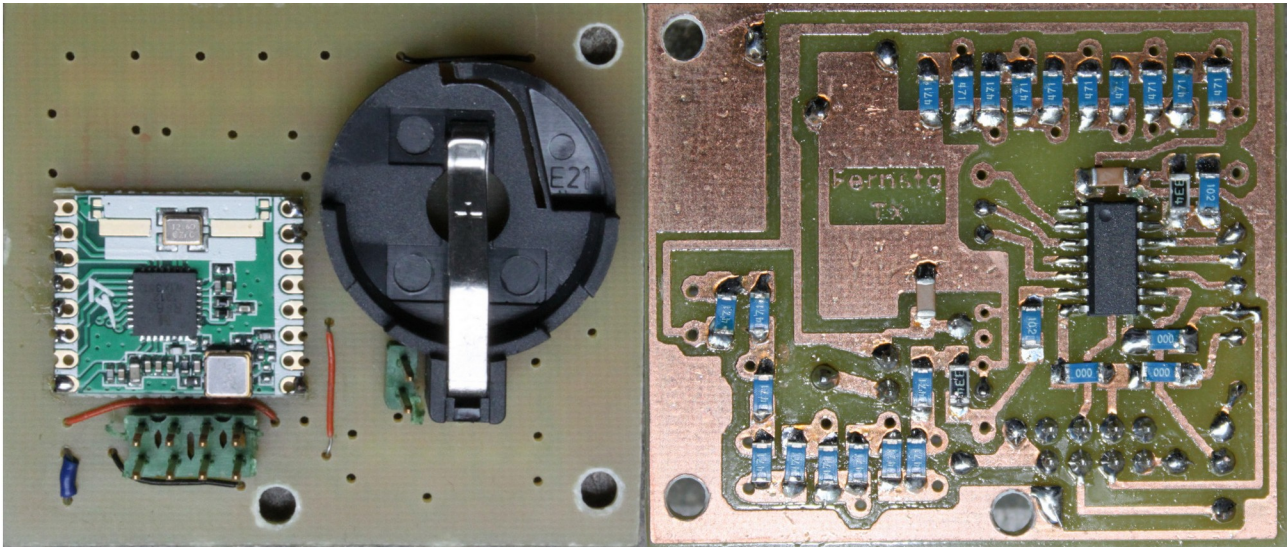


Bild 20: Aufbau der Platinen – hier die Senderplatine mit Vorder- und Rückseite

Bei den Übertragungsversuchen mit den neuen Aufbauten zeigte sich ein sonderbares Verhalten. Nach dem Einschalten der Versorgungsspannung blieb der Empfänger in 8 von 10 Fällen taub – er reagierte auf keine Signale. Auf einer der IRQ-Leitungen war in den Fällen ein 20kHz Rechtecksignal zu sehen.

Ich hatte deshalb einen logischen Kurzschluss vermutet, also einen Ablauf in dem Software und Fernsteuermodul sich zu so einem Fehlerbild aufschaukeln. In der Folge hatte ich die Software Stück für Stück abgespeckt bis nur noch der Teil für die Parametrierung des Moduls übrig war. Das Problem bestand trotzdem weiter. Schließlich habe ich die Software so erweitert, dass ich die Parametrierschritte einzeln von Hand auslösen konnte. Da war dann zu sehen, dass die wilden Schwingungen mit dem letzten Parametrierbefehl - Übergang von STANDBY in EMPFANG begann. Nach langer Suche habe ich schließlich eine Abhilfe gefunden: Die Parametrierung einfach wiederholen. Auf diese Weise konnte der Fehler (diesmal eindeutig auf der Seite des Fernsteuermoduls) behoben werden.

5.3 Ein- und Ausgaben sind um einen Tastendruck versetzt

Das dritte Problem bei dem es viel Zeit gebraucht hat bis es behoben war hing mit dem FIFO und dessen Verhalten zusammen. Für die korrekte Funktion meiner Applikation war es wichtig, dass eine gedrückte Taste sofort erkannt wurde und nicht erst nachdem die Information durch den FIFO gelaufen war.

Wie oben schon beschrieben wurde das eigentliche Problem dadurch umgangen, dass ein Tastendruck nicht nur einmal sondern so oft gesendet wird, dass der gesamte FIFO mit der neuen Information gefüllt wird. Voraussetzung dafür dass das funktioniert ist aber, dass der FIFO zuvor komplett ausgelesen wurde. Steht noch alte Info im FIFO werden nur so viele neue Telegramme empfangen bis der FIFO gefüllt ist, die überzähligen Telegramme werden einfach ignoriert. Kommt wegen irgendwelcher Störungen nur ein Teil der gesendeten Telegramme am Empfänger an, sieht man nach der Übertragung nicht die aktuelle Taste sondern immer noch die alte. Nach einem solchen „Fehler“ bleibt diese Verschiebung um eine Taste erhalten.

Es hat eine Weile gebraucht bis auch dafür eine Lösung gefunden war: Wenn der FIFO leer ist und man versucht ihn weiter auszulesen, wird die letzten Information die in ihn eingeschrieben worden war ausgegeben. Man kann also sicherstellen, dass der FIFO immer leer ist wenn man ihn mindestens so oft ausliest wie er lang ist. Damit sollte das Problem der um eins versetzten Tasten eigentlich gelöst sein. Trotzdem trat der Fehler weiterhin auf. Längere Versuchsreihen haben dann gezeigt, dass es, wenn er schon beim ersten Tastendruck auftritt, bleibt oder er tritt beim ersten Tastendruck nicht auf, dann bleiben die Tastenbedienungen auch bei allen weiteren Bedienungen synchron. Kurz – das Verhalten hing davon ab, wie der Anlauf verlaufen war.

Letztendlich hat sich nach vielen Versuchen herausgestellt, dass nicht aus Auslesen aus dem Fernsteuermodul das Problem war sondern der Fehler sich erst viel später beim Weiterreichen der Information zu dem nächsten Mikrocontroller eingeschlichen hat. Also auch in diesem Fall ein Fehler in der eigenen Software.

6. Eine überraschende Zahl

Wie oben bei den technischen Daten beschrieben, hat der Sender einen Pegel von maximal 12dBm, der Empfänger bei 25kb/s eine Empfindlichkeit von -111dBm. Geht man von einem Aufbau aus, bei dem die Antennen jeweils 0dB Gewinn haben, kommt man auf einen Abstand von 123dB. Sucht man sich die Formel für die Freiraumdämpfung raus (z.B. Rothammel Kapitel 2.9.1) kommt man auf

$$\frac{P_s}{P_e} = \left(\frac{4 \pi R}{\lambda} \right)^2;$$

stellt man die Formel etwas um, ergibt sich

$$R = \frac{\lambda}{4 \pi} \sqrt{\frac{P_s}{P_e}}; \text{ mit } \lambda = 0,346 \text{ m und } \frac{P_s}{P_e} = 10^{12,3}; \text{ ergibt sich}$$

$$R = \frac{0,346 \text{ m}}{4 \pi} * \sqrt{10^{12,3}} = 38,9 \text{ km};$$

Ohne Störungen durch andere Sender und ohne Hindernisse würde man damit also eine Strecke von knapp 39 km überbrücken können. In dem realen Aufbau liegen die Antennengewinne natürlich deutlich unter 0dB, zwischen Sender und Empfänger werden sich auch immer Hindernisse befinden. Der 868MHz-Bereich ist auch von anderen Nutzern belegt, was die Reichweite auch stark einschränken kann. Wenn sich dadurch auch nur ein Bruchteil der theoretisch erreichbaren Übertragungsstrecke nutzen lässt ist es doch eine ganz beachtliche Zahl.

7. Zusammenfassung

In den letzten Jahren hat sich der Trend verstärkt dass Module bei stark sinkenden Preisen immer schneller immer komplexer werden. Das hier vorgestellte Funkmodul ist nur ein Beispiel davon. Die geringen Preise laden fast schon dazu ein solche Module dann in eigenen Projekten einzusetzen. Was man dabei gerne unterschätzt, ist der Aufwand den man treiben muss damit ein mit solchen Modulen aufgebautes Gerät genau das tut, was man sich vorgestellt hat. Rechnet man den Zeitaufwand den man treiben muss bis man am Ziel ist, ist es vielleicht günstiger, mehr Geld hinzulegen und gleich ein komplettes und funktionierendes Gerät zu kaufen. Auf der anderen Seite ist der Lerneffekt nicht zu unterschätzen, den man erzielt wenn man sich bis zum

Erfahrungen mit einer aktuellen Fernsteuerung

erwünschten Ziel durchkämpft.

8. Quellenverzeichnis

Hoperf Electronic Datenblatt RFM63W V1.2

Rothammels Antennenbuch 13. Auflage ISBN 978-3-88692-065-5 DARC Verlag