# nRF24L01+ 2.4 GHz Sniffing eines HS220 Remote Controls

Fakultät der Angewandten Informatik
der Hochschule für Technik und Wirtschaft Berlin

## Projektdokumentation

Im Fach Drahtlose Netzwerke
Herr Prof. Dr. Huhn

vorgelegt von

**Nikita Shevchenko**
geboren am 15.07.1998

im Februar 2023

htw.
Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

## Einleitung

Im Rahmen des Kurses Drahtlose Netzwerke an der Hochschule für Technik und Wirtschaft Berlin ist für jeden Studierenden die Vorgabe, eine Projektarbeit inkl. Dokumentation anzufertigen. Die Projektarbeit wurde von Prof. Dr. Alexander Huhn vorgeschlagen und beinhaltete zuerst das Reverse Engineering eines Remote Controls einer HS220 Drohne. Im Laufe der Projektarbeit wurde das Thema jedoch weiter eingeschränkt. Diese Dokumentation widerspiegelt somit die Hard- und Software-Komponente der Projektarbeit sowie das Vorgehen zum Sniffing eines proprietären Protokolls im 2.4-GHz-Bereich einer Fernsteuerung einer Holy Stone 220 Drohne.

## Motivation

Eine Aussage über die Sicherheit eines Systems zu treffen, ist meistens schwierig. Alle Komponente und dessen Komplexität spielen eine gravierende Rolle in der Einschätzung. Ein Remote Control kann aus erster Sicht viele mögliche Sicherheitslücken umfassen, doch die Frage "Wie und mit welchen Mitteln sich diese entdecken lassen", ist unklar. Diese Ungewissheit stellt eine einladende Herausforderung dar.

## Zielsetzung

Das Ziel dieser Projektarbeit ist es, die Datenpakete eines Remote Controls einer Drohne im 2.4-GHz-Bereich mithilfe eines nRF24L01+ Moduls, das an ein Arduino UNO angeschlossen ist, zu empfangen.

# Inhaltsverzeichnis

# 1 Hardware-Komponente

Im Folgendem werden die in dieser Projektarbeit benutzte Hardware-Komponente näher beschrieben. Zuerst werden die Hardware-Komponente der Drohne bzw. der Fernsteuerung betrachtet. Als Nächstes wird die Auswahl der passenden Hardware für das Sniffing näher beschrieben. Es werden die Funktionalitäten der Hardware für das Sniffing erläutert. Letztens wird für das Projekt verwendetes Arduino UNO Board kurz beschrieben.

## 1.1 Hardware der Fernsteuerung

Nach der Zerlegung der Fernsteuerung der Drohne wurde festgestellt, dass die Steuerung der Drohne mithilfe eines XNS1042CV Mikroprozessors umgesetzt wird. Der eingebettete Mikroprozessor beinhaltet einen Hochfrequenz-Transceiver-Chip XN297 mit der Funktion des Frequency Hoppings. Dieser operiert in einem Teil des ISM-Bandes, nämlich 2.4 – 2.483 GHz und unterstützt Datenraten von einem und zwei Megabyte pro Sekunde.

## 1.2 Auswahl der Hardware für das Sniffing

Der Frequenzbereich der Fernsteuerung wird für WLAN nach IEEE 802.11b, 802.11g, 802.11n sowie ZigBee und Bluetooth verwendet. Im selben Frequenzbereich können auch proprietäre Protokolle für die drahtlose Kommunikation eingesetzt werden. Für das Sniffing können SDR (Software Defined Radio) sowie drahtlose Netzwerkadapter zur Anwendung kommen. SDRs werden angewendet, um drahtlose Kommunikation auch im 2.4-GHz-Bereich zu analysieren. Einige Nachteile dieser Hardware beinhalten Komplexität und Inflexibilität der zugehörigen Software sowie hohe Kosten für das Gerät. Drahtlose Netzwerkadapter, welche über ein Monitor-Mode verfügen, können eingesetzt werden, um WLAN-Pakete bei 2.4-GHz zu empfangen, um diese mithilfe einer Software wie z. B. Wireshark zu untersuchen. Wesentlicher Nachteil ist die Möglichkeit der HW nur WLAN-Pakete in bestimmten Frequenzbereich zu empfangen. Im Laufe der Internet-Recherche wurde festgestellt, dass der HF-Chip XN297 mithilfe eines Klons, nämlich nRF24L01+ emuliert werden kann. Das nRF24L01+ Board kann man für das nicht explizit vorgesehene Sniffing im Frequenzbereich von 2.4 – 2.483 GHz verwenden.

## 1.3 nRF24L01+

nRF24L01+ ist ein Radio-Transceiver Modul. nRF24L01 Chip wird oft im IoT-Bereich angewendet, da die Hardware sehr kostengünstig ist und eine hohe Flexibilität durch bereits

vorhandene Bibliotheken anbietet. nRF24L01+ verfügt über eine SPI-Schnittstelle, welche für die Steuerung des Moduls vorgesehen ist.

## 1.4 Sniffing mit nRF24L01+

Wie bereits erwähnt, ist der nRF24L01+ Modul nicht explizit für das Sniffing vorgesehen. Jedoch basierend auf der Erforschung von Travis Goodspeed kann man nRF24L01+ für das Sniffing konfigurieren. Durch das Ersetzen der Receiver-MAC-Adresse durch die XN297-Chip-Preamble wird der nRF24L01+ Modul verwirrt, sodass er nicht mehr nur das Payload zurückliefert (Normalfall), sondern beinhaltet jetzt auch die MAC-Adresse des Paketempfängers. Die CRC-Prüfsummen müssen ebenfalls ausgeschaltet werden. Dies ermöglicht es alle mögliche Datenpakete zu empfangen, jedoch kann es gleichzeitig dabei auch zu fehlerhaften Daten führen. Die eingestellte Datenpräambel als RX-Adresse lässt das Modul so reagieren, dass es die Datenpräambel als MAC-Adresse bei ankommenden Datenpaketen identifiziert und bei übereinstimmender Datenpräambel durch diese aktiviert wird.

## 1.5 Arduino UNO

Ein Arduino UNO mit einem ATmega328P Mikroprozessor wird verwendet, um das nRF24L01+ Radio Modul über eine SPI-Schnittstelle zu manipulieren. Die Software wird im nächsten Kapitel erläutert.

# 2 Software-Komponente

Das nRF24L01+ Modul wird über die SPI-Schnittstelle an ein Arduino UNO Board angeschlossen. Die Programmierung der Software für das Sniffing erfolgt über Arduino IDE und wird mittels Sprache C umgesetzt. Es werden bereits existierende Bibliotheken für das Radio-Modul verwendet.

## 2.1 Arduino IDE

Arduino IDE wird als Basis für die Entwicklung des Projekts verwendet. Die Entwicklungsumgebung ermöglicht es, Arduino UNO über ein USB-Kabel zu programmieren.

## 2.2 RF24

RF24 [**?**]article) - ein OSI-Layer 2 Treiber bzw. eine Bibliothek für das nRF24L01+ Modul. Diese wird verwendet, um das Modul zu steuern.

## 2.3 Projektcode

Arduino UNO ist folgendermaßen für das Projekt programmiert: Das nRF24L01+ Modul wird in der Methode scan sowie setupRadio (neu)konfiguriert:

Auto acknowledgement wird ausgeschaltet, Power Amplifier wird zu RF24_PA_MIN gesetzt, Datenrate wird auf 1 Megabyte pro Sekunde gesetzt, die Größe des Payloads wird auf 32 Bytes gesetzt, die Startfrequenz für das Scannen wird gesetzt.

```
1 radio.setAutoAck(false);
2 radio.setPALevel(RF24_PA_MIN);
3 radio.setDataRate(DATA_RATE);
4 radio.setPayloadSize(32);
5 radio.setChannel(channel);
```

Die Länge der Adresse wird auf 3 Bytes gesetzt. Das Modul eröffnet eine Pipe zum Lesen und verwendet dabei anstatt der MAC-Adresse des Empfängers die Präambel des Senders.

```
1 radio.setAddressWidth(ADDR_WIDTH);
2 radio.openReadingPipe(1, kbPipe);
```

Cyclic Redundancy Check (CRC) wird ausgeschaltet und das Radio wird für das Empfangen der Daten gestartet.

```
radio.disableCRC();
radio.startListening();
```

Mithilfe einer Endlosschleife, welche für die Kontrolle des Frequenzkanals zuständig ist, werden Datenpakete empfangen: Dabei befindet sich der ausgewählte Frequenzkanal zwischen 30. und 70. Kanal d. h. im Bereich von 2.430 - 2.470-GHz.

```
if (channel > 70)
   channel = 30;

sp("Tuning to ");
Serial.println(2400 + channel);
radio.setChannel(channel++);
```

Der auf einem Frequenzkanal verbrachte zeitliche Abstand wird mit 10.000 Millisekunden beschränkt.

```
time = millis();
while (millis() - time < wait)
{...}
```

Sobald das Radio Bytes zum Ablesen hat, wird das Payload bzw. die MAC-Adresse des Empfängers und das Payload in ein Buffer mit entsprechender Größe gespeichert.

```
if (radio.available()) {
   radio.read(&p, PKT_SIZE);
   ... }
```

Der Buffer wird in Hexadezimalzahlen umgewandelt und an einen seriellen Anschluss ausgegeben.

```
for(int i=0; i < PKT_SIZE; i++)
{
   printHex(p[i]);
}
```

# 3 Ergebnisse der Projektarbeit

Durch das Steuern der Drohne mithilfe der Fernsteuerung bzw. eines Remote Controls werden Daten drahtlos übertragen. Diese Kommunikation wird mittels eines nRF24L01+ Moduls mitgehört. Es wurde nur ein Joystick während des Sniffings betätigt. Die empfangenen Daten werden ausgegeben.

```
Tuning to 2433
AA 23 67 B1 1C FC E5 46 0D AE 88 88 12 69 CE 69
AA 27 67 B1 3C FD ED 66 0D AE AD 98 36 69 EE EB
AA 23 67 B1 3D FD A5 66 1D 2E 8C 88 12 69 EE 69
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 23 67 B1 3C FC E5 6E 0D AE 8C 88 16 6A EE 69
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 3B 42 EB
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 67 31 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA A3 67 B1 1C FC E5 66 0D AE 8C 08 12 69 EE 69
AA 23 67 B1 1D FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 47 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 63 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 67 B1 1C CA 5D EA 5B 2A 08 E0 9E 39 42 EB
AA 67 67 B1 1C FC E5 66 1D BE ED 0B 92 69 ED E9
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 67 B1 1C CA C4 EA 5B 2A 08 E0 9E 39 42 EB
AA 67 67 B1 1C FD E5 66 0D AE 8C A8 12 E9 EE A9
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 6B
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 6B EE 69
AA 22 C7 B1 1D F9 E5 67 1D AE AC 88 EC BD 5A C5
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 BE 3B 42 EB
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 23 6F B1 1C CA 44 EA 5B 6A 09 E0 9E 7B 42 EB
AA 22 67 B1 1C FC E5 66 0D AE 8C 88 12 69 EE 69
AA 47 6F B3 5C FC E9 6E 8D AE 9D 98 16 6B EE 6B
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 23 67 B1 3C FC E5 66 0D AE 8C 88 12 6B EE 69
AA 23 67 B1 1C CA 44 EA 5B 2A 08 E0 9E 39 42 EB
AA 23 67 B1 1C FC E5 66 0D AE 8C 88 32 69 EE 69
```

Es werden keine Daten empfangen, wenn die Fernsteuerung nicht betätigt wird. Dabei

können fehlerhafte Datenpakete durch das Rauschen verursacht werden.

Die ersten 8 Bit (AA) deuten auf die letzten 8 Bit der Präambel hin. Die systematisch wiederkehrenden Bitfolgen veranschaulichen variable Kommunikation (variable Betätigung des Joysticks) des Remote Controls mit der Drohne.

Das XN297-Chip kann Scrambling-Algorithmen verwenden. Diese Algorithmen werden mit hoher Wahrscheinlichkeit auch bei der Kommunikation der Fernsteuerung mit der Drohne angewendet. Das Payload muss entschlüsselt werden, um die Wiederherstellung der Kommunikation bzw. Datenübertragung zu gewährleisten.

# 4 Quellenverzeichnis

[1] Candell, R., Hany, M., Lee, K. B., Liu,Y., Quimby, J., Remley, K. (2018): April Guide to Industrial Wireless Systems Deployments, URL: https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.300-4.pdf [09.02.2023]
[2] nRF24 Treiber / Bibliothek (seit 2012): URL: https://github.com/nRF24/RF24 [09.02.2023]
[3] Panchip (2014): XN297-Datasheet, URL: https://datasheetspdf.com/datasheet/XN297.html [09.02.2023]
[4] Travis Goodspeed (2011): "Promiscuity is the nRF24L01+'s Duty", URL: https://travisgoodspeed.blogspot.com/2011/02/promiscuity-is-nrf24l01s-duty.html [09.02.2023]
[5] Samy Kamkar (2015): KEYSWEEPER, URL: https://samy.pl/keysweeper/ [09.02.2023]
[6] pascallanger (seit 2016): DIY-Multiprotocol-TX-Module, URL: https://github.com/pascallanger/DIY-Multiprotocol-TX-Module [09.02.2023]