SVEUČILIŠTE U SPLITU

FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

**SIGURNOST BEŽIČNIH MREŽA**

**Hakiranje bežičnog zvona upotrebom Arduina i HackRF-a**

Anica Vuković

Siječanj 2021.

Sadržaj

[1 UVOD 1](#_Toc61557287)

[2 OPREMA (HARDVER) 3](#_Toc61557288)

[2.1 Arduino Uno 3](#_Toc61557289)

[2.2 HackRF One 3](#_Toc61557290)

[2.3 434MHz ASK RF Transmitter 4](#_Toc61557291)

[2.4 Bežično zvono 5](#_Toc61557292)

[3 SOFTVER 6](#_Toc61557293)

[3.1 SDRSharp 6](#_Toc61557294)

[3.2 Audacity 6](#_Toc61557295)

[3.3 Arduino Software (IDE) 7](#_Toc61557296)

[3.4 Universal Radio Hacker 8](#_Toc61557297)

[4 Postupak izrade projekta 9](#_Toc61557298)

[4.1 Određivanje frekvencije zvona i snimanje signala 9](#_Toc61557299)

[4.2 Analiza signala 11](#_Toc61557300)

[4.3 Replay napad 14](#_Toc61557301)

[4.3.1 Arduino replay napad 14](#_Toc61557302)

[4.3.2 HackRF replay napad 19](#_Toc61557303)

[5 Zaključak 22](#_Toc61557304)

[6 Literatura 23](#_Toc61557305)

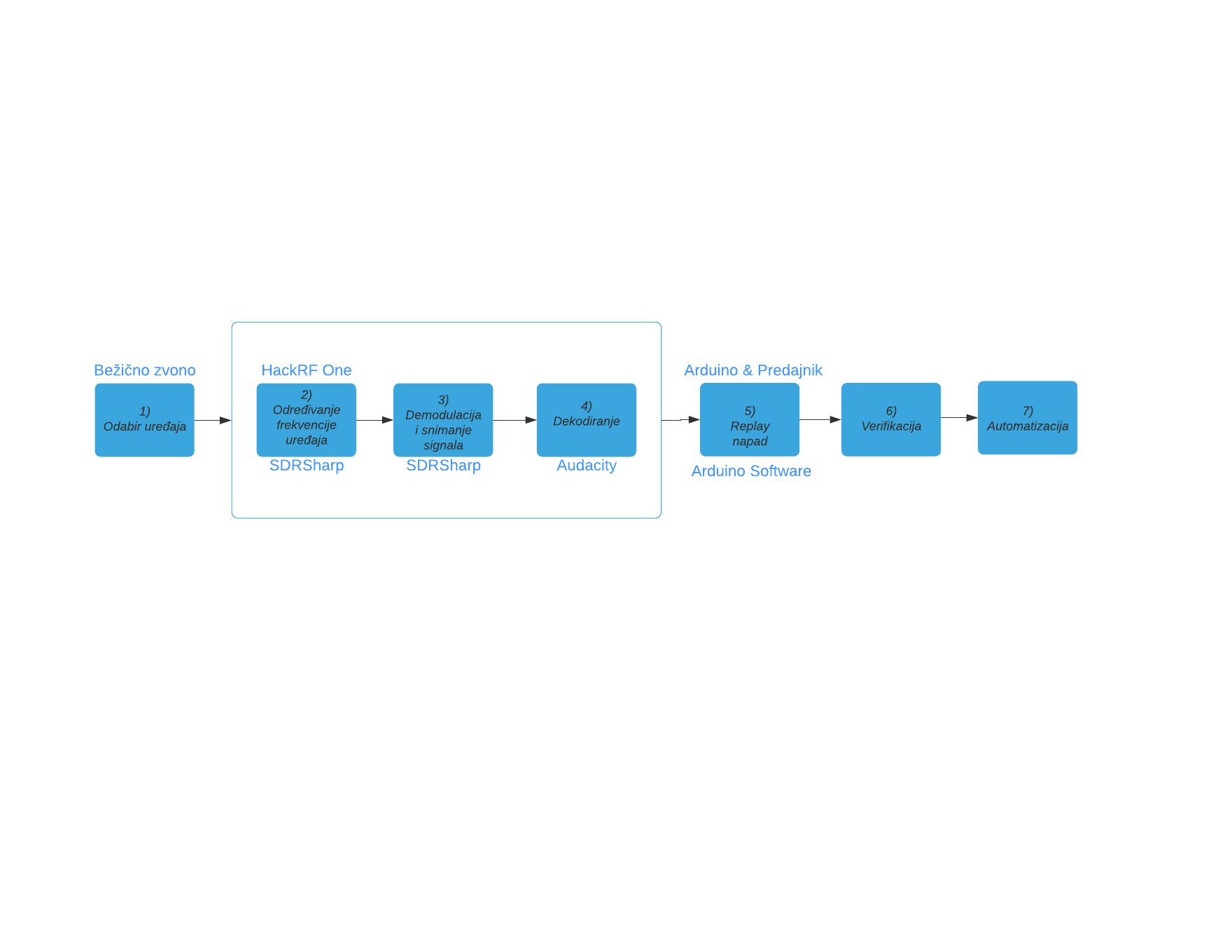
# UVOD

Bežična komunikacija jedno je od područja koje raste velikom brzinom i čije tehnologije postaju sve popularnije u svakodnevnom životu. Sa sve većom količinom informacija koje se šalju putem bežične mreže, pojavljuju se i nove prijetnje.

Veliki je broj uređaja za bežičnu komunikaciju koji koriste radio valove ili infracrveno svjetlo kako bi ostvarili dvosmjernu komunikaciju. Upotrebom softverski definiranih radiouređaja (SDR), čija se cijena kreće već od 20ak dolara, moguće je dobro istražiti područje radio komunikacija. Zahvaljujući softverski definiranim radiouređajima, koji rade na frekvenciji između 30 MHz i 6000 MHz, moguće je poremetiti rad svih uređaja koji rade unutar tog frekvencijskog opsega.

Uređaji poput bežičnih ključeva za automobile, alarmnih sustava, sustava za uzbunu, televizora, prekidača, zvona su uglavnom vrlo loše konfigurirani te ne implementiraju nikakve aspekte zaštite komunikacije i iz tog razloga su podložni replay napadima.

U ovom radu prikazan je napad na jedan takav uređaj, bežično zvono za vrata. Metodologija napada prikazana je na slici 1.1.



Slika 1.1 - Metodologija napada

Odabrani uređaj je bežično zvono. Na početku je potrebno odrediti frekvenciju na kojoj zvono radi. Danas, uglavnom, takvi jednostavni uređaji rade na frekvencijama od 315 MHz, 430 MHz ili 900 MHz. U ovom projektu korišten je program SDRSharp kako bi se odredila frekvencija na kojoj zvono radi. Korištenjem SDRSharp-a s priključenim HackRF uređajem uočena je veza između signala i pritiska na gumb u SDRSharp-u na frekvenciji od 433,9 MHz te je jasno da uređaj radi na toj frekvenciji.

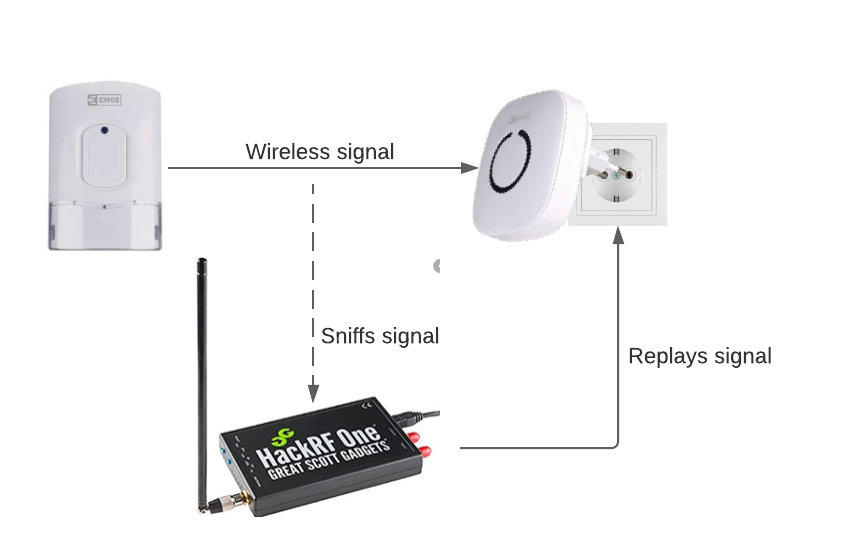
U SDRSharp-u je snimljen signal na način da se snimala pojava signala na toj frekvenciji prilikom nekoliko pritisaka na gumb. Snimljeni signal je spremljen (WAV format) i za njegovo dekodiranje je korišten program Audacity.

Signal ima amplitudnu modulaciju koja je najjednostavniji oblik digitalne modulacije signala. Binarni podaci su opisani različitim razinama amplitude. Primjerice, ako je vrijednost amplitude različita od 0, znači da je primljena informacija 1 dok je u suprotnom 0. Takva modulacija je značajno olakšala interpretaciju signala u Audacity-u.

Na temelju analize napravljene u Audacity-u (određena su razdoblja i trajanje nula i jedinica) napisan je kod .

Budući da je poznato da je signal na frekvenciji od 433,9 MHz te da ima amplitudnu modulaciju, uzet je jednostavni ASK odašiljač koji radi na toj frekvenciji te se pomoću njega i Arduina izvršio replay napad. Zvono je uspješno zazvonilo što znači da je napad uspio.

Osim toga, korišten je i program Universal Radio Hacker u kojem je signal snimljen i ponovno reproduciran te je na taj način izvršen „replay attack“ prikazan na slici 1.2.



Slika 1.2 – Replay napad

# OPREMA (HARDVER)

U nastavku je prikazana oprema korištena prilikom izrade ovog projekta.

## Arduino Uno

Arduino je sjajna platforma za razvoj hardvera i softvera te omogućuje jako brzu izradu hardvera. Dakle, da bi Arduino obavio neku funkciju, mora se programirati, odnosno napisati program koristeći Arduino IDE. Kako bi se napisani programi mogli prebaciti u mikrokontroler na razvojnoj pločici koristi se USB veza s računalom. Arduino je open-source te se njegova cijena kreće oko 20ak dolara. Za potrebe ovog projekta korišten je Arduino UNO prikazan na slici 2.1. [1]

Slika na kojoj se prikazuje tekst, elektronički, strujni krug

Opis je automatski generiran

Slika 2.1 - Arduino UNO [2]

## HackRF One

HackRF One tvrtke Great Scott Gadgets softverski je definirana radio periferna jedinica sposobna za prijenos ili prijam radio signala od 1 MHz do 6 GHz. Dizajniran kako bi omogućio testiranje i razvoj suvremenih radio tehnologija i sljedeće generacije, HackRF One je hardverska platforma otvorenog koda koja se može koristiti kao USB periferna jedinica ili programirati za samostalni rad. HackRF One prikazan je na slici 2.2. [3]

Slika na kojoj se prikazuje tekst, elektronički, prilagodnik

Opis je automatski generiran

Slika 2.2 - HackRF One [4]

## 434MHz ASK RF Transmitter

443MHz RF je povoljan i pristupačan standard za RF komunikaciju između dva uređaja često korištena za daljinsko upravljanje, bežičnu komunikaciju i slanje podataka. Važno je imati na umu da ovaj uređaj u potpunosti ovisi o frekvenciji i modulaciji uređaja na koji pokušavate prenijeti. [5] Odašiljač je nabavljen tek nakon što je utvrđeno da zvono koristi radiofrekvenciju od ~ 434MHz i da je digitalna modulacija ASK. Predajnik korišten za ovaj projekt prikazan je na slici 2.3, a kupljen je u Chipoteci po cijeni od 20,90 kn.[[1]](#footnote-1)

Slika na kojoj se prikazuje elektronički, strujni krug

Opis je automatski generiran

Slika 2.3 – Prijemnik i predajnik

## Bežično zvono

Za ovaj projekt korišteno je bežično zvono Emos dometa 120 metara, uz 52 melodije te koje radi na frekvenciji od 433,92MHz. Korišteno zvono prikazano je na slici 2.4., a kupljeno je u Chipoteci po cijeni od 134,00 kn. [[2]](#footnote-2)

Slika na kojoj se prikazuje elektronički

Opis je automatski generiran

Slika 2.4 - Bežično zvono EMOS

# SOFTVER

U nastavku će biti opisani programi korišteni tijekom izrade projekta.

## SDRSharp

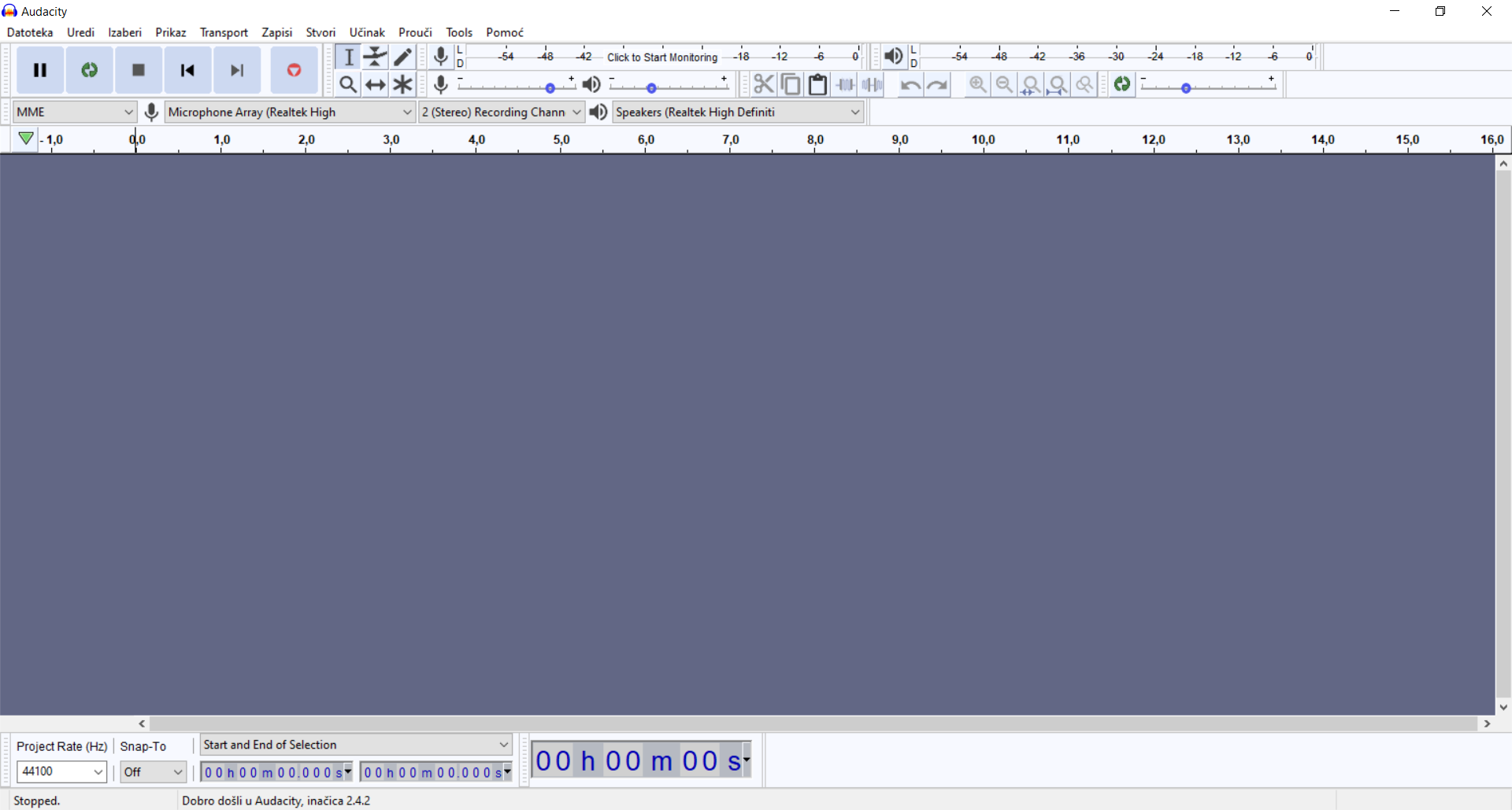
SDRSharp je program koji omogućuje GUI sučelja za vizualizaciju i preslušavanje signala putem SDR uređaja. Sučelje programa prikazano je na slici 3.1.



Slika 3.1 - SDRSharp

## Audacity

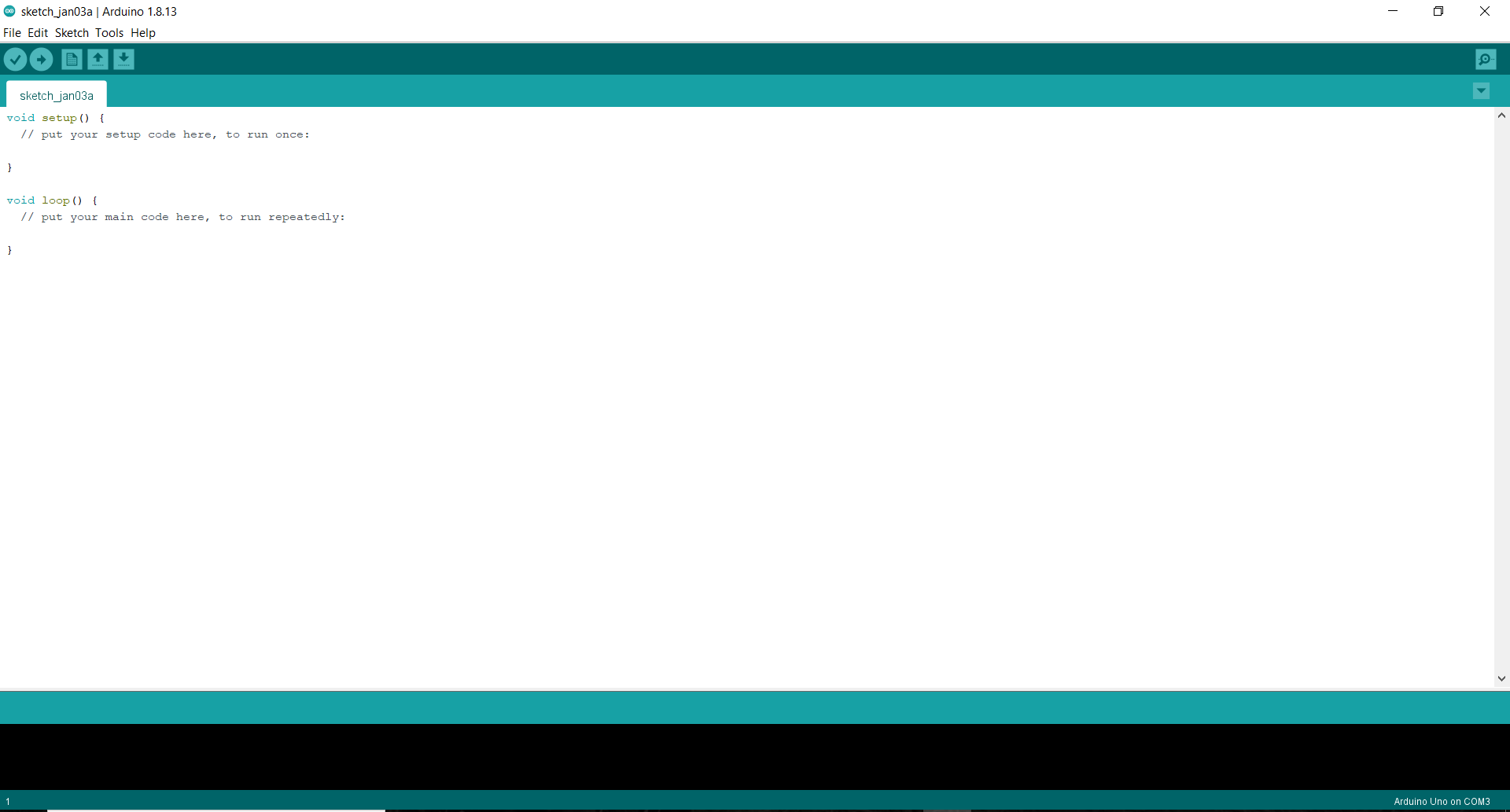
Audacity je besplatna aplikacija za modificiranje audio datoteka. Koristimo ga za promatranje i tumačenje radio signala. Sučelje programa prikazano je na slici 3.2.



Slika 3.2 - Audacity

## Arduino Software (IDE)

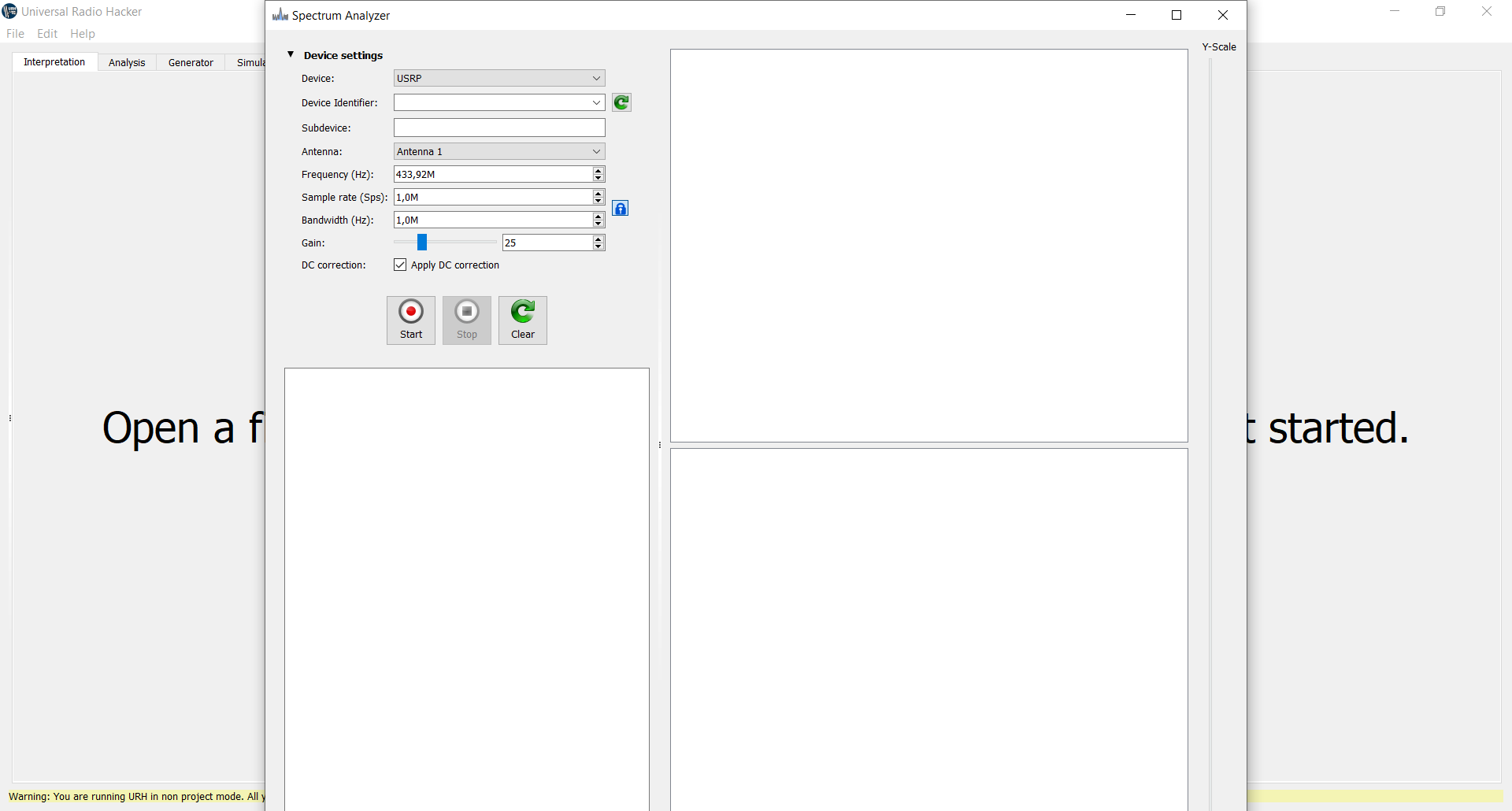
Arduino softver koji je open-source (IDE) olakšava pisanje koda i prijenos na ploču. Ovaj se softver može koristiti s bilo kojom Arduino pločom. Sučelje programa prikazano je na slici 3.3.



Slika 3.3 - Arduino INO

## Universal Radio Hacker

Universal Radio Hacker (URH) alat je za analizu nepoznatih bežičnih protokola. Pomoću tog programa može se snimiti te „replayati“ signal. Sučelje programa prikazano je na slici 3.4.



Slika 3.4 - Universal Radio Hacker

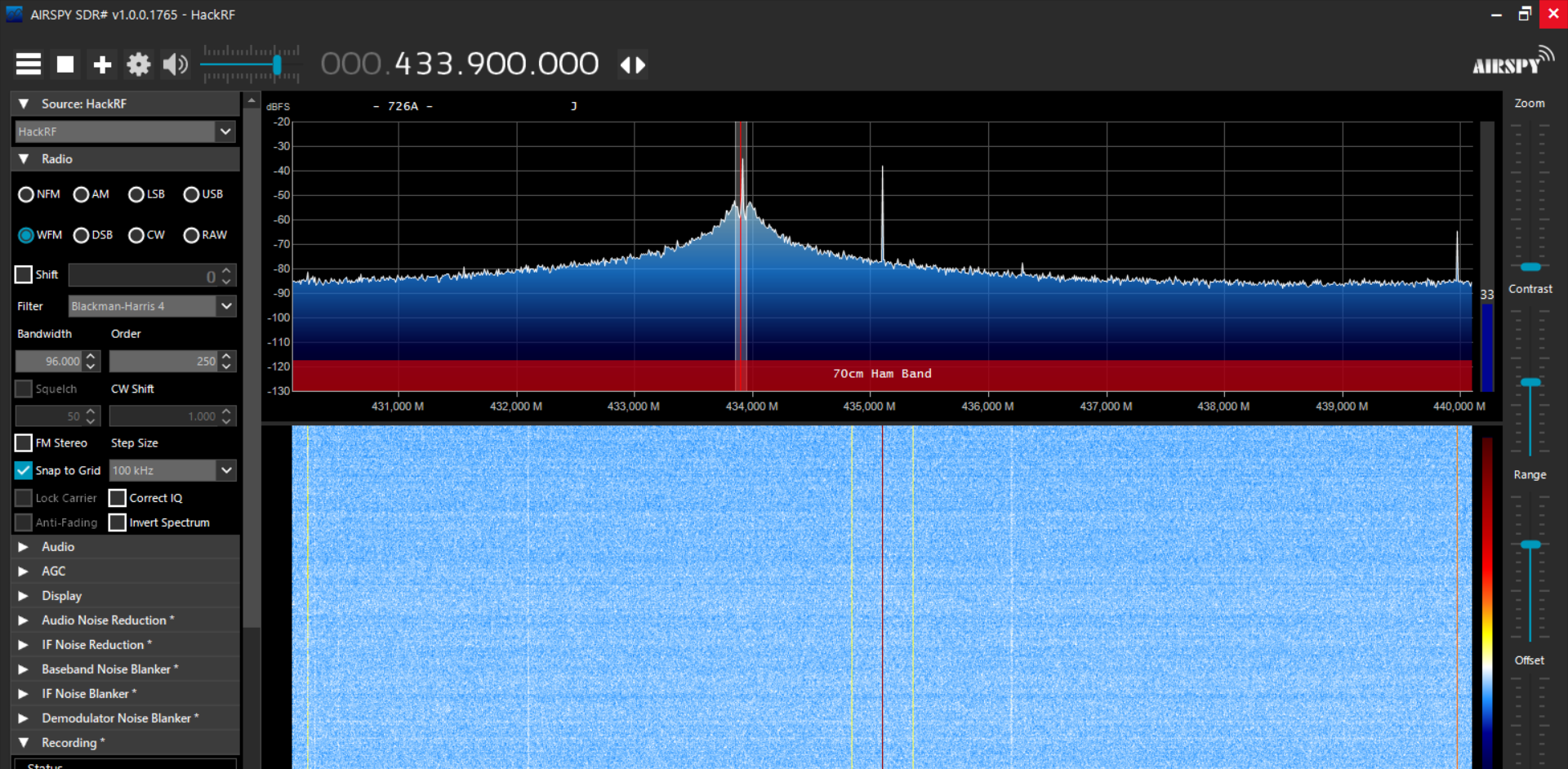
# Postupak izrade projekta

U nastavku je opisan detaljan postupak izrade projekta.

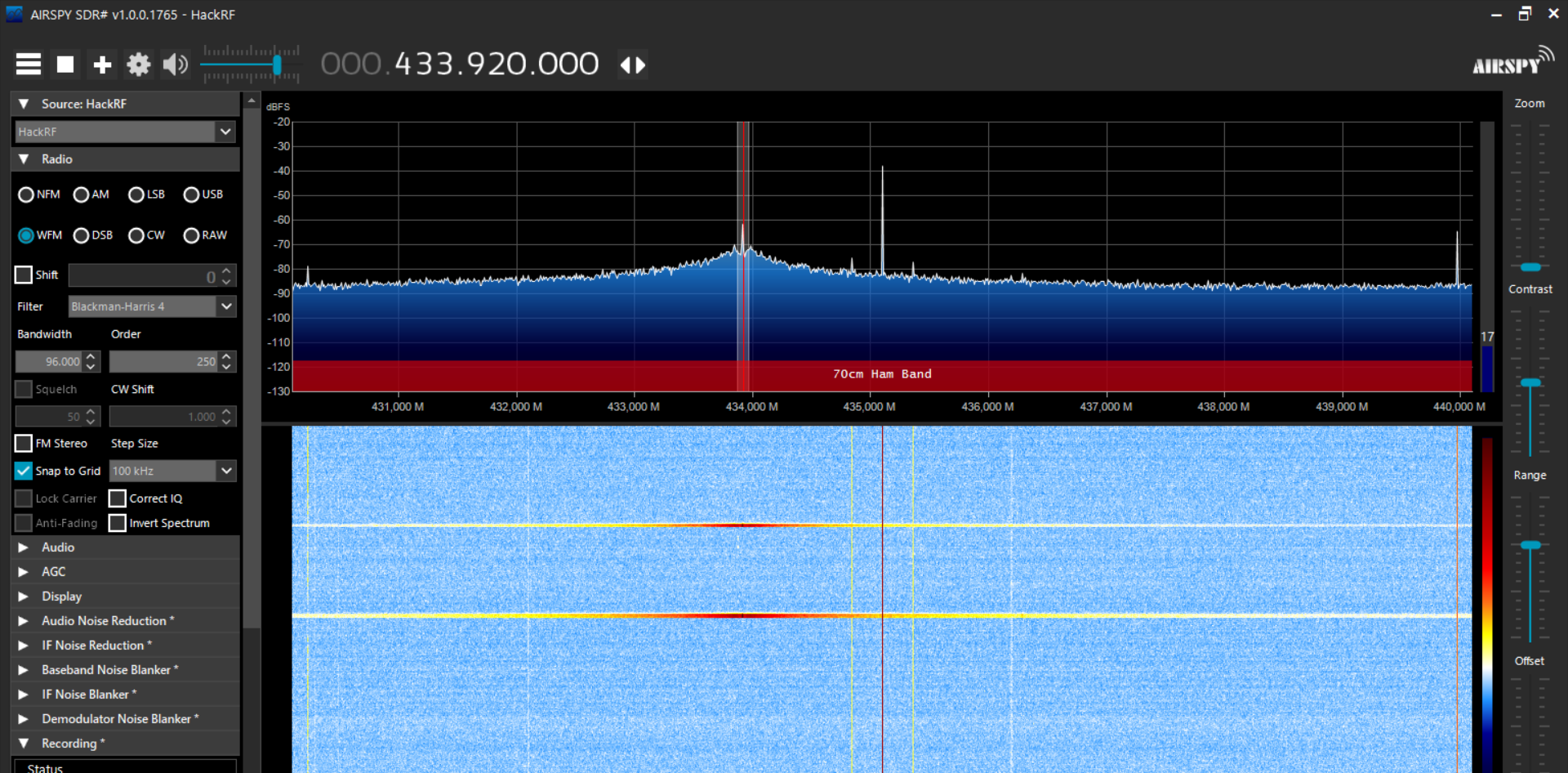
## Određivanje frekvencije zvona i snimanje signala

Kao što je već napisano, kako bi se odredila frekvencija na kojoj radi bežično zvono korišten je program SDRSharp te uređaj HackRF One.

Kako bi pravilno odredili frekvenciju, potrebno je prvo postaviti sve parametre u SDRSharp-u. Potrebno je kao izvor staviti SDR uređaj koji se koristi, a u ovom slučaju to je bio HackRF One. Važno je LNA i VGA Gain postaviti na vrijednost veću od 0, odnosno povećati ih na neku veću vrijednost. U programu možemo namještati frekvenciju na kojoj želimo promatrati pomoću „Frequency Inputa“. Polje za unos frekvencije podijeljeno je na 4 dijela od kojih svaki sadrži 3 vrijednosti. Prvi dio predstavlja vrijednosti u GHz, drugi u MHz, treći u kHz te četvrti u Hz (npr. 000.000.000.000). Na primjer, za radio stanicu na 88.6 MHz upišemo 000.088.600.000 u polje za unos frekvencije. Budući da su poznate frekvencije na kojima rade jednostavni uređaji poput zvona krenulo se sa slušanjem na frekvenciji od 300 MHz (000.300.000.000 ).   
Glavni dio prozora vizualno prikazuje RF spektar kao graf u stvarnom vremenu. Aktivni signali pojavit će se kao vrhovi na ovom grafikonu. Pritiskom gumba na našem zvonu nije vidljiva pojava nikakvih signala te se iz tog razloga pomiče na veću frekvenciju. Uočena je veza između signala i pritiska na gumb u SDRSharp-u na frekvenciji od 433,9 MHz te je vidljivo da uređaj radi na toj frekvenciji. Svakim pritiskom na gumb u SDRSharp-u se na frekvenciji od 433,92 MHz pojavljuje se signal kao vrh na grafikonu što je vidljivo naslici 4.1islici 4.2**.**

****

Slika 4.1 - Signal u SDRSharp-u

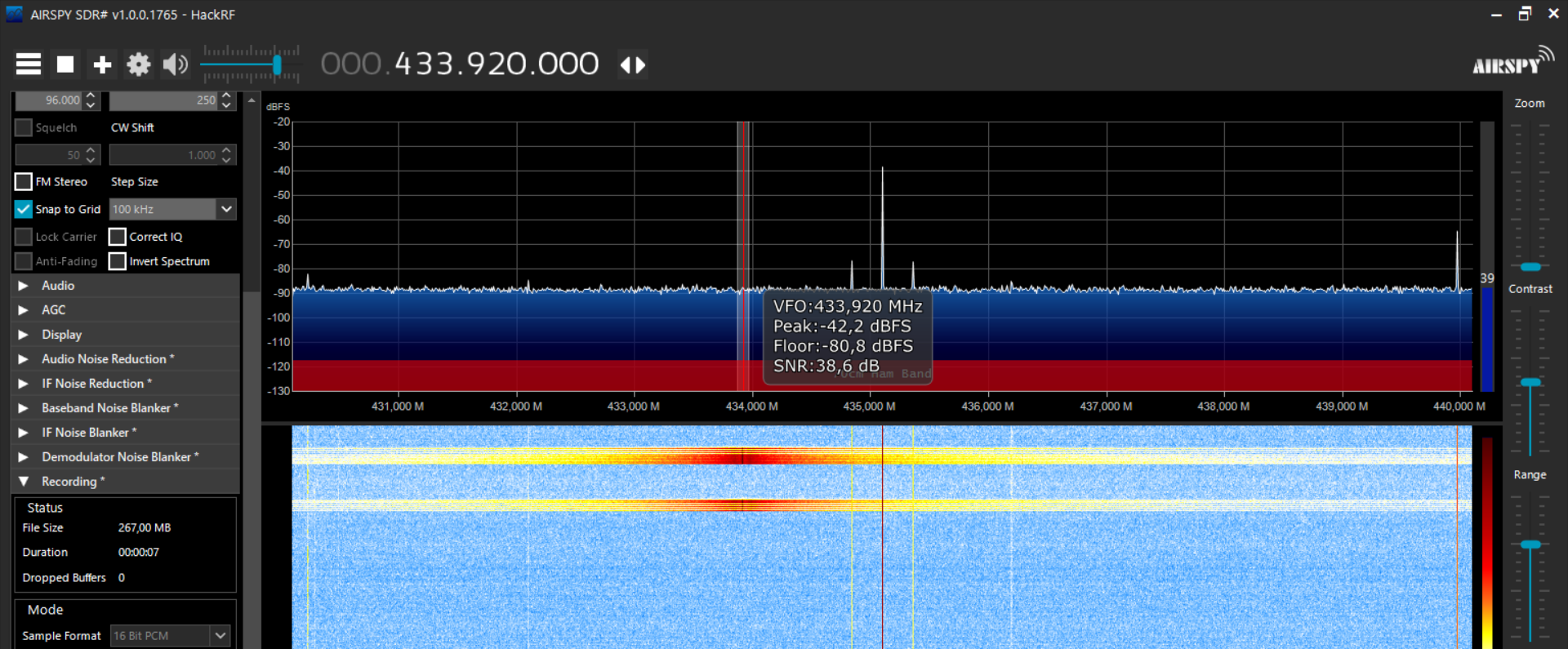
****

Slika 4.2 - Signal u SDRSharp-u

Na taj način je uspješno određena frekvencija na kojoj zvono radi.

Nakon određivanja frekvencije potrebno je snimiti signal. U SDRSharpu kartica za snimanje omogućuje snimanje I / Q i audio snimaka.

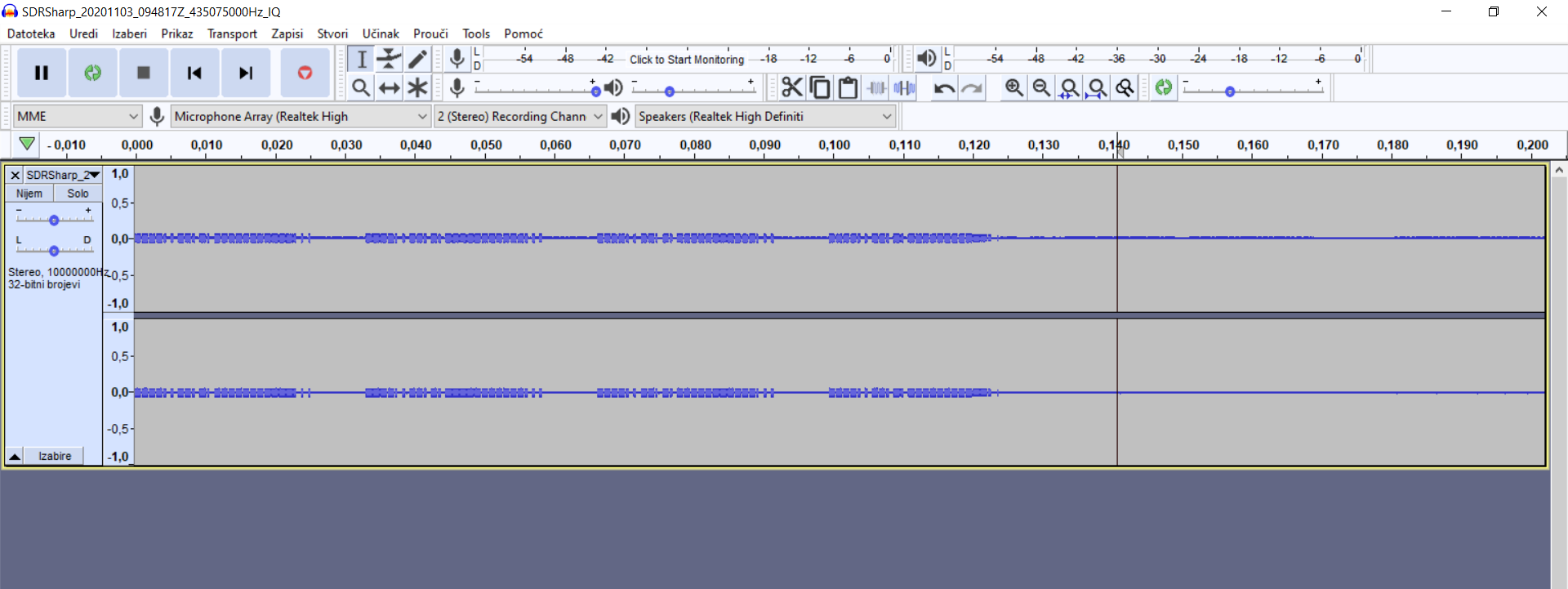
I / Q snimka je snimka cijele širine pojasa ~ 2 MHz na koju je trenutno podešeno. Sprema RF podatke unutar širine pojasa, tako da se mogu kasnije reproducirati. I / Q snimanje može se izvršiti potvrđivanjem okvira "Baseband". I / Q snimke mogu se reproducirati u SDR # odabirom IQ datoteke (\* .wav) s izbornika izvora. Na taj način je snimljen signal bežičnog zvona tako da je snimljen signal koji se pojavljuje prilikom nekoliko pritisaka na dugme. Snimanje signala je prikazano na slici 4.3. Dobivena datoteka na se koristila za analizu signala.

****

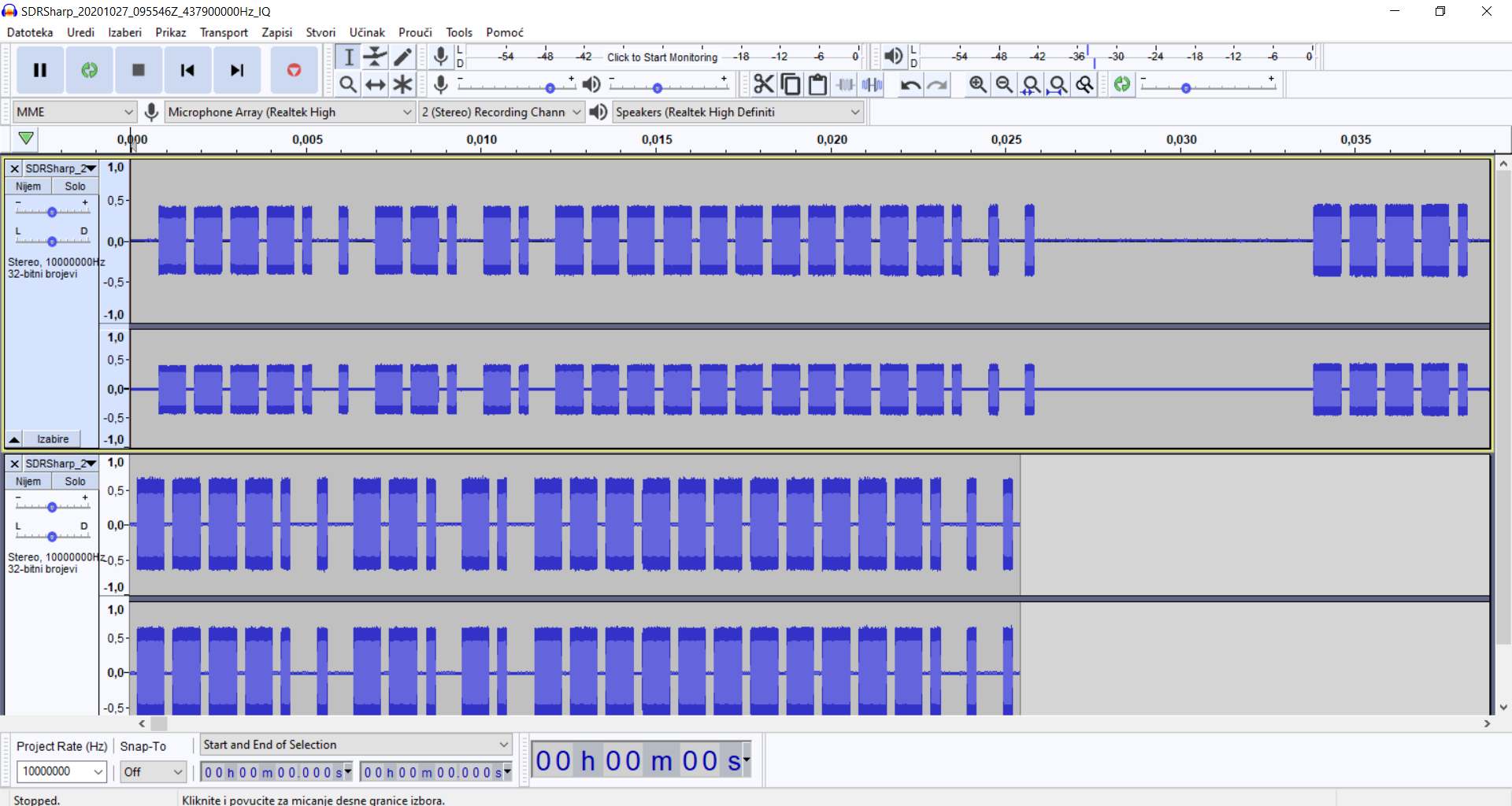
Slika 4.3 - Snimanje signala u SDRSharp-u

## Analiza signala

Nakon što je signal snimljen, analiziran je u programu Audacity. Promatrajući signal u Audacity-u vidi se da se jednim pritiskom na gumb pojavljuju 4 jednaka intervala što je prikazano naslici 4.4, a svaki od tih intervala je sastavljen od 25 pod intervala što se vidi naslici 4.5. Potrebno je odrediti vrstu modulacije. Najčešće korištene digitalne modulacije su diskretna modulacija amplitude sinusnog signala, diskretna modulacija frekvencije sinusnog signala te diskretna modulacija faze sinusnog signala. Analizom signala u Audacity-u zaključujemo da signal ima amplitudnu modulaciju. Amplitudna modulacija jednostavno modulira amplitudu signala za slanje podataka. Kod ASK amplituda signala nosioca se mijenja prema modulacijskom signalu na način da signal postoji kada se prenosi simbol '1' odnosno ne postoji kada se prenosi '0', dok su frekvencija i amplituda signala nositelja konstantne. To je prikazano na slici 4.6.



Slika 4.4 - Analiza signala u Audacity-u



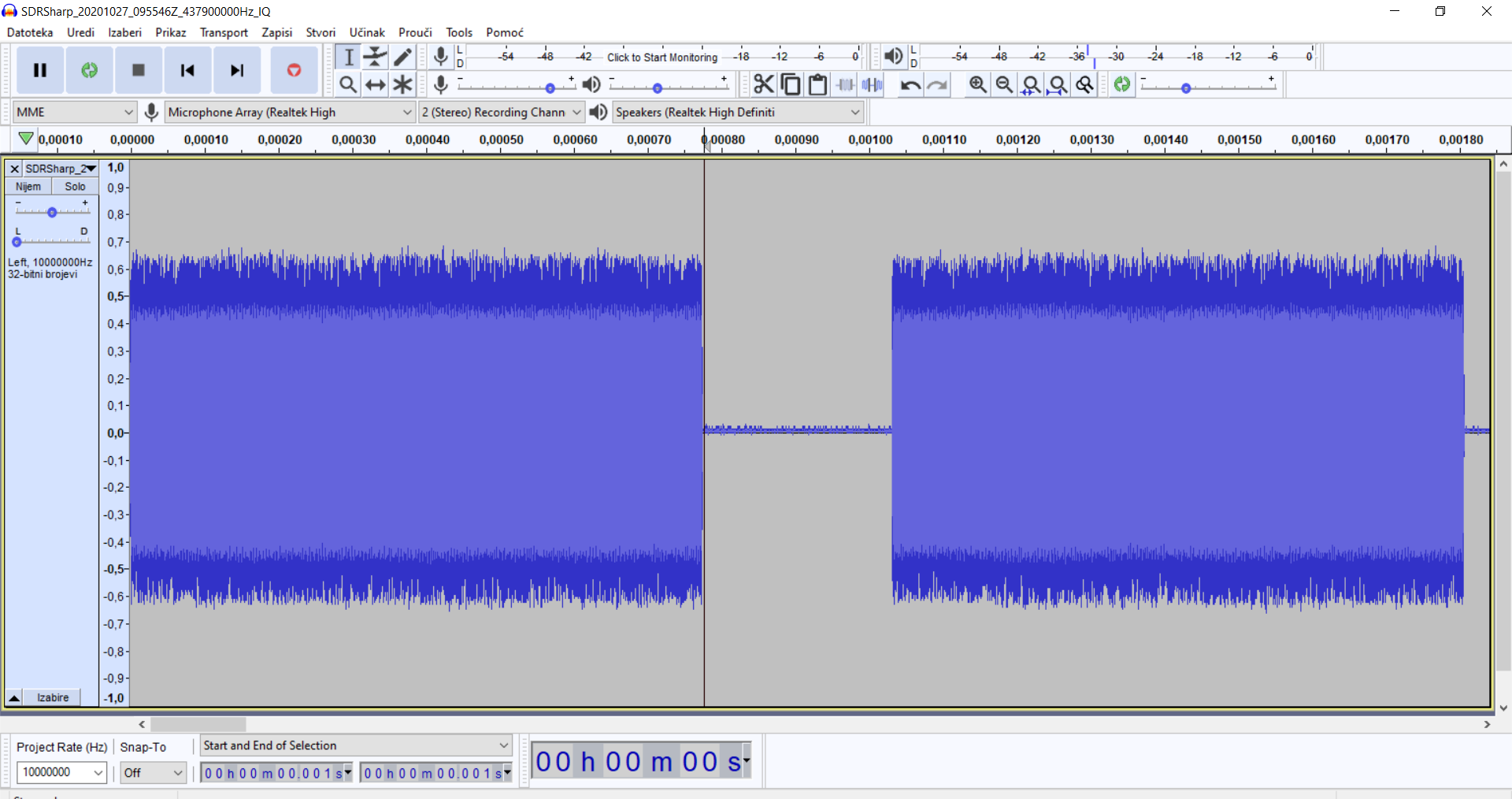
Slika 4.5 - Analiza signala u Audacity-u

Slika na kojoj se prikazuje stol

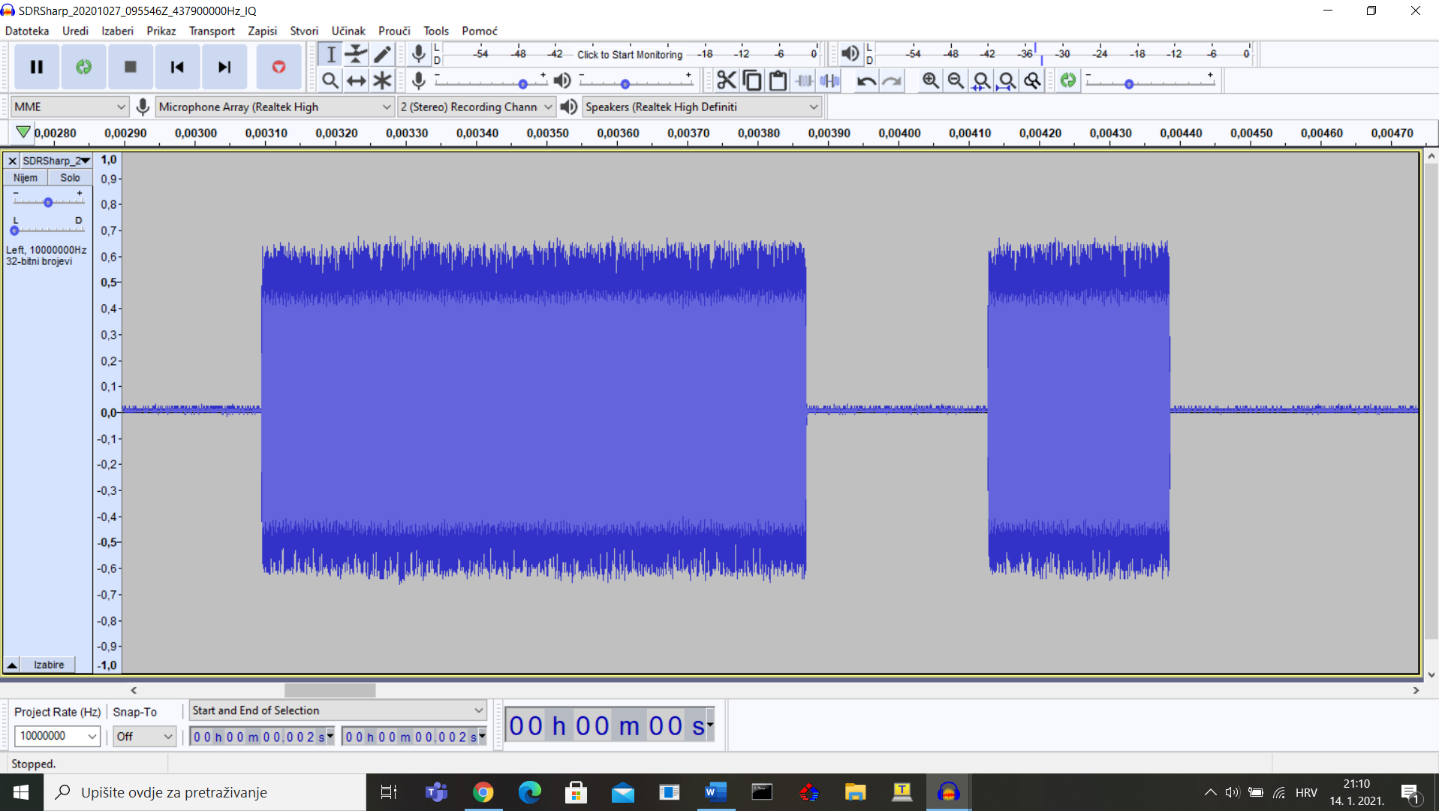
Opis je automatski generiran

Slika 4.6 - Analiza signala u Audacity-u

Kada je amplituda 0, to će se interpretirati kao 0, a kada je veća to će se slati kao 1. Na prethodnoj slici se vidi da postoje 2 različita perioda trajanja jedinice. Kako bi odredili trajanje svake jedinice potrebno je zumirati signal na slici 4.6 te dobijemo sliku 4.7. Zumiranjem slike 4.7 vidimo da prva jedinica traje do 0,000780 sekundi što je 780 mikro sekundi. Sa iste slike odredi se trajanje druge jedinice oduzimanjem krajnje i početne točke (0,001807-0,001030) te se dobije 0,000777 što je približno 780 mikro sekundi te zaključujemo da prve dvije jedinice jednako traju. Na isti način (oduzimanjem krajnje i početne točke) odredimo trajanje ostalih jedinica te dobijemo da traju ili 780 ili 260 mikro sekundi što se vidi na slici 4.8.



Slika 4.7 - Analiza signala u Audacity-u



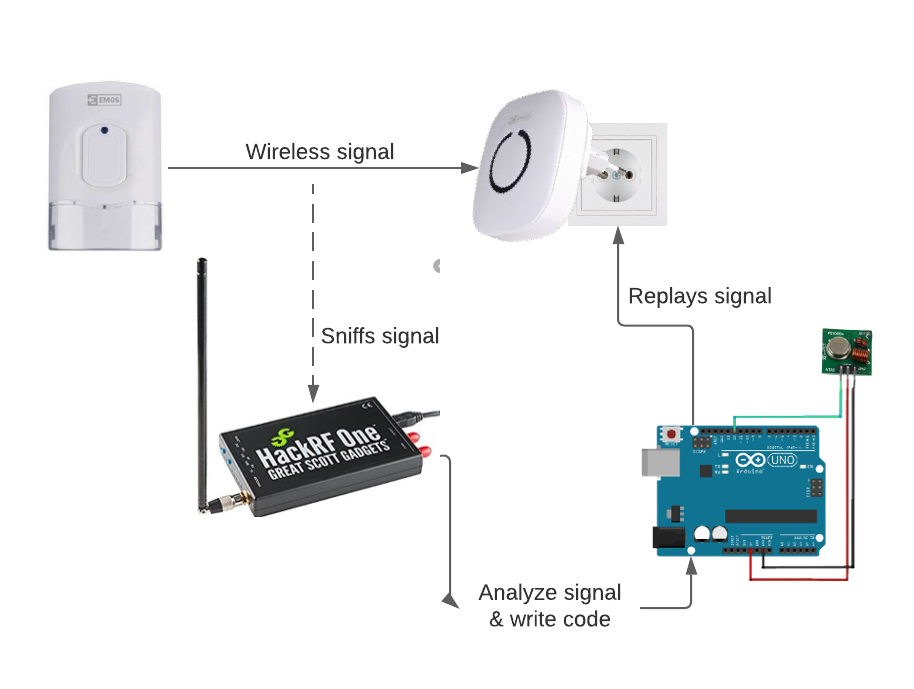
Slika 4.8 - Analiza signala u Audacity-u

## Replay napad

U nastavku je opisano izvođenje Arduino replay napada te HackRF replay napada.

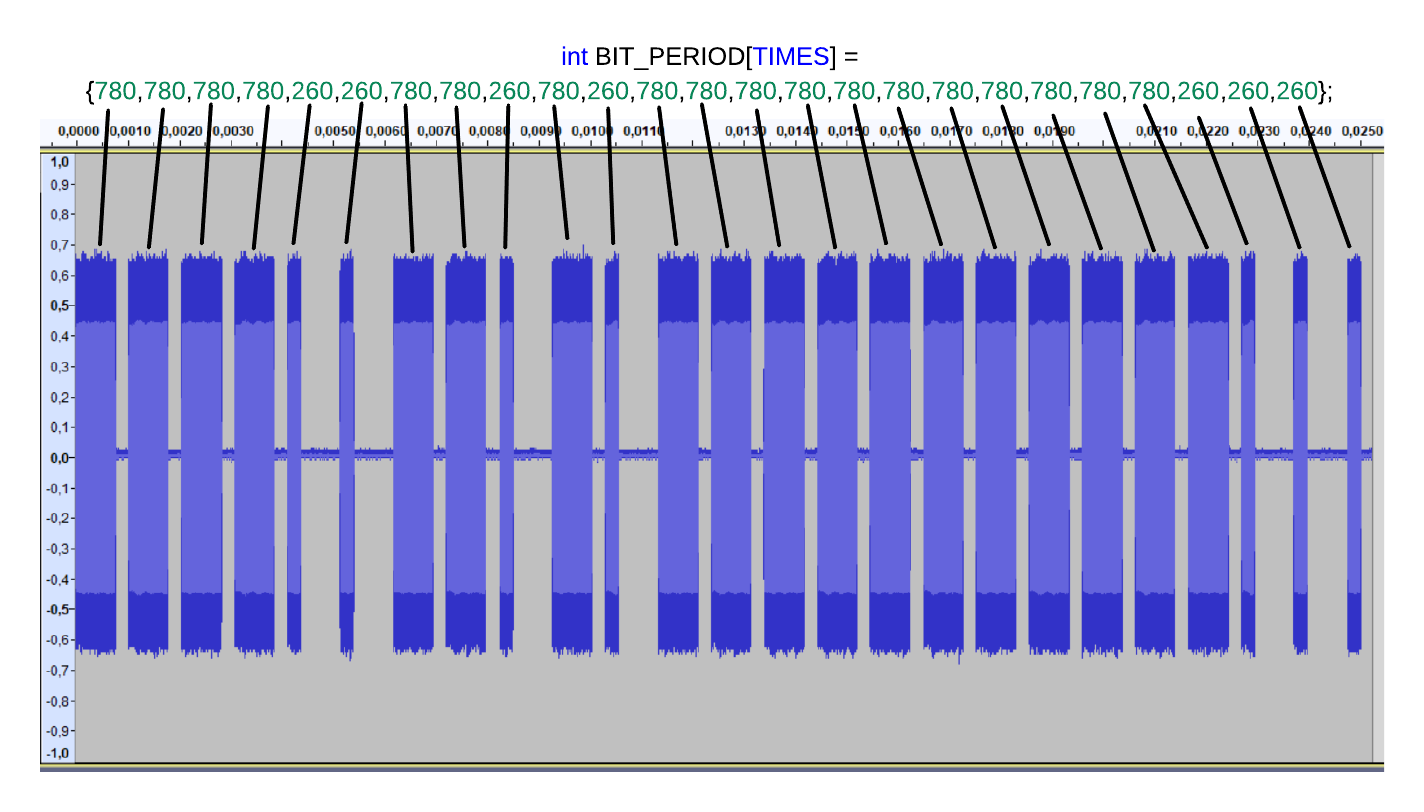
### Arduino replay napad

Kako bi se uspješno izveo Arduino replay napad prikazan na slici 4.9 potrebno je napisati kod koji će se pokrenuti na Arduino pločici te spojiti Arduino i predajnik.



Slika 4.9 - Arduino replay napad

Na temelju analize signala potrebno je zapisati trajanje svake jedinice u niz koji se zove BIT\_PERIOD. Na slici 4.10 vidi se koliko traje svaka jedinica.



Slika 4.10 - Trajanje podintervala

Na taj način je poznato trajanje svake jedinice, ali važno je znati u kojem trenutku jedinica i počinje, odnosno u kojem trenutku nula završava. Iz tog razloga u niz times spremi se početak svake jedinice. Prva jedinica počinje u 0, nakon toga druga jedinica počinje u trenutku 0.00104 te tako dalje zumiranjem signala odredimo početak svake jedinice kao na slici 4.11.

Slika na kojoj se prikazuje tekst, snimka zaslona, izlog, elektronički

Opis je automatski generiran

Slika 4.11 - Niz "times"

U nastavku je prikazan cijeli kod:

#define TX\_PIN 9

#define BIT\_PERIOD1 780

#define BIT\_PERIOD2 260

#define TIMES 25

float times[TIMES] = {

 .0000, .00104, .002075, .003112,  .004148, .005184, .00622, .007257,

 .008294, .009329, .010366, .011402,  .012438, .013475, .014511, .015548,

 .016584, .017621, .018656, .019694,  .02073, .021767, .02284, .02384, .024876

};

int BIT\_PERIOD[TIMES]={ 780,780,780,780,260,260, 780,780,260,780,260,780,780,780,780,780,780,780,780,780,780,780,260,260,260};

/\* .0079 je razmak za 4 ova \*/

/\*trajanje prva cetiri je 0.000780 to je 780ms \*/

/\*peti je 0,000258 tj 260ms\*/

/\*sesti 260ms\*/

/\* 7,8 isto 780, 9. 260, 10. 780, 11. 260

12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 traju 780

23,24,25 po 260 \*/

void setup()

{

  pinMode(TX\_PIN, OUTPUT);

  Serial.begin(115200);

  Serial.println("");

  Serial.println(F("Hacking Wireless Doorbell"));

  Serial.println(F("by Anica Vukovic"));

}

void ring\_bell()

{

  Serial.println("Ringing!\n");

  // zvono šalje 4 jednaka intervala pritiskom na gumb pa je zato for petlja kroz koju se prolazi 4 puta

  for (int j = 0; j < 4; j++)

  {

    single\_ring();

  }

}

void single\_ring()

{

  int last = 0;

  for (int i = 0; i < TIMES; i++)

  {

    int us = times[i] \* 1000000;

    if (i != 0)

      delayMicroseconds(us - last - BIT\_PERIOD[i-1]);

    // pošalji jedinicu za BIT\_PERIOD koji je ili 780 ili 260 mikro sekundi

    digitalWrite(TX\_PIN, HIGH);

    delayMicroseconds(BIT\_PERIOD[i]);

    digitalWrite(TX\_PIN, LOW);

    last = us;

  }

  delay(20);

}

void loop()

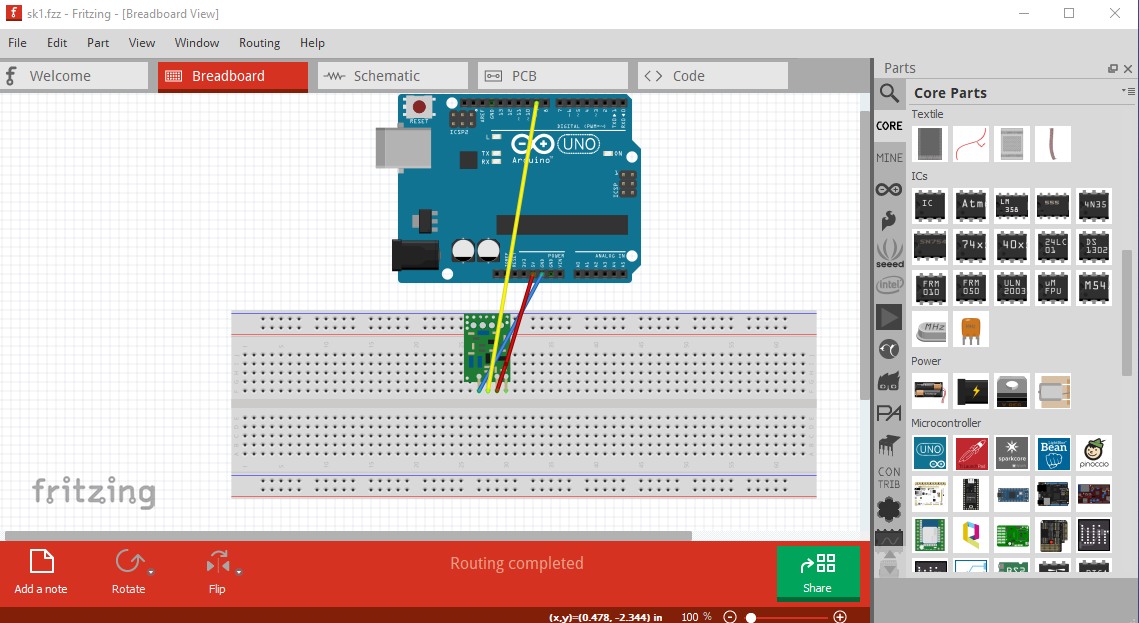
{

  ring\_bell();

  delay(30000);

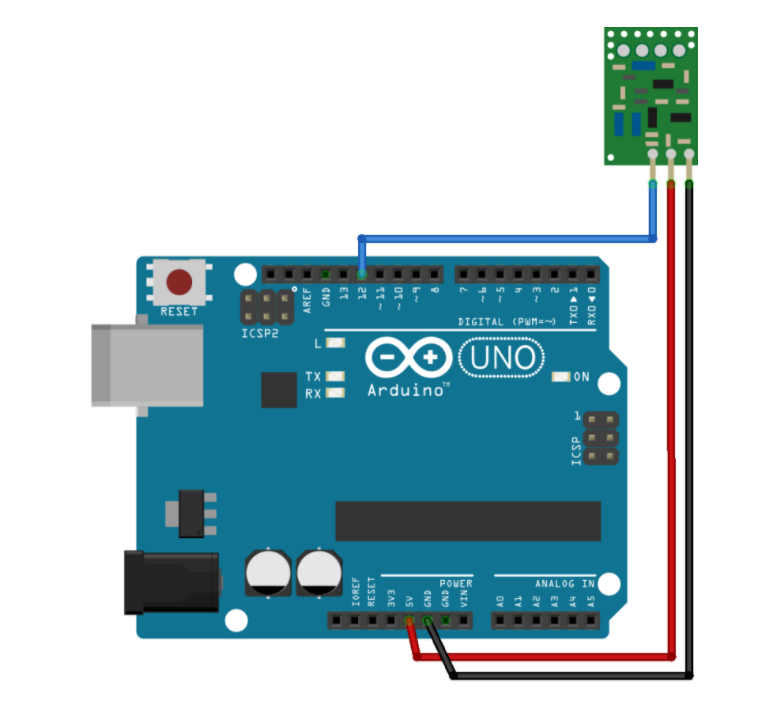
}

Shema spajanja Arduina i predajnika prikazana je naslici 4.12.



Slika 4.12 - Spajanje Arduina i predajnika

Shema je bolje vidljiva na slici 4.13

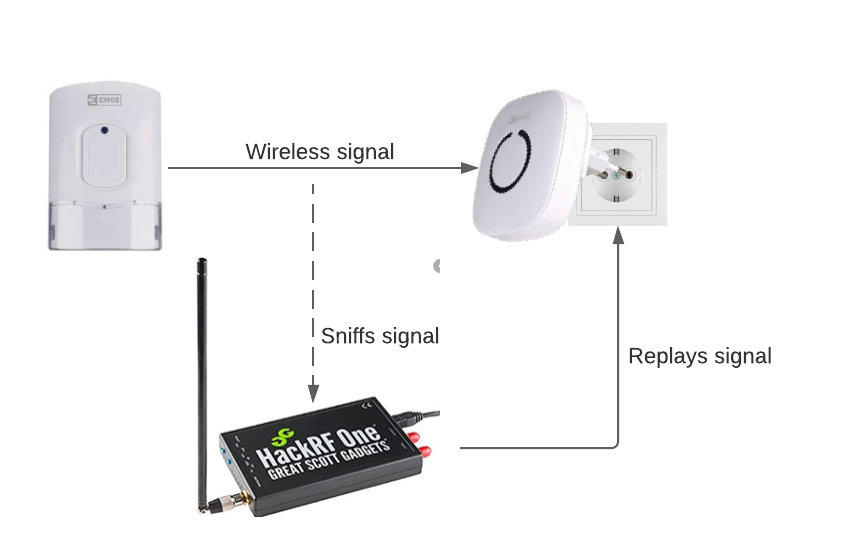


Slika 4.13 - Spajanje Arduina i predajnika [7]

Nakon što je kod učitan na Arduino ploču i pokrenut, uspješno je zvonilo zvono što znači da je replay napad uspio.

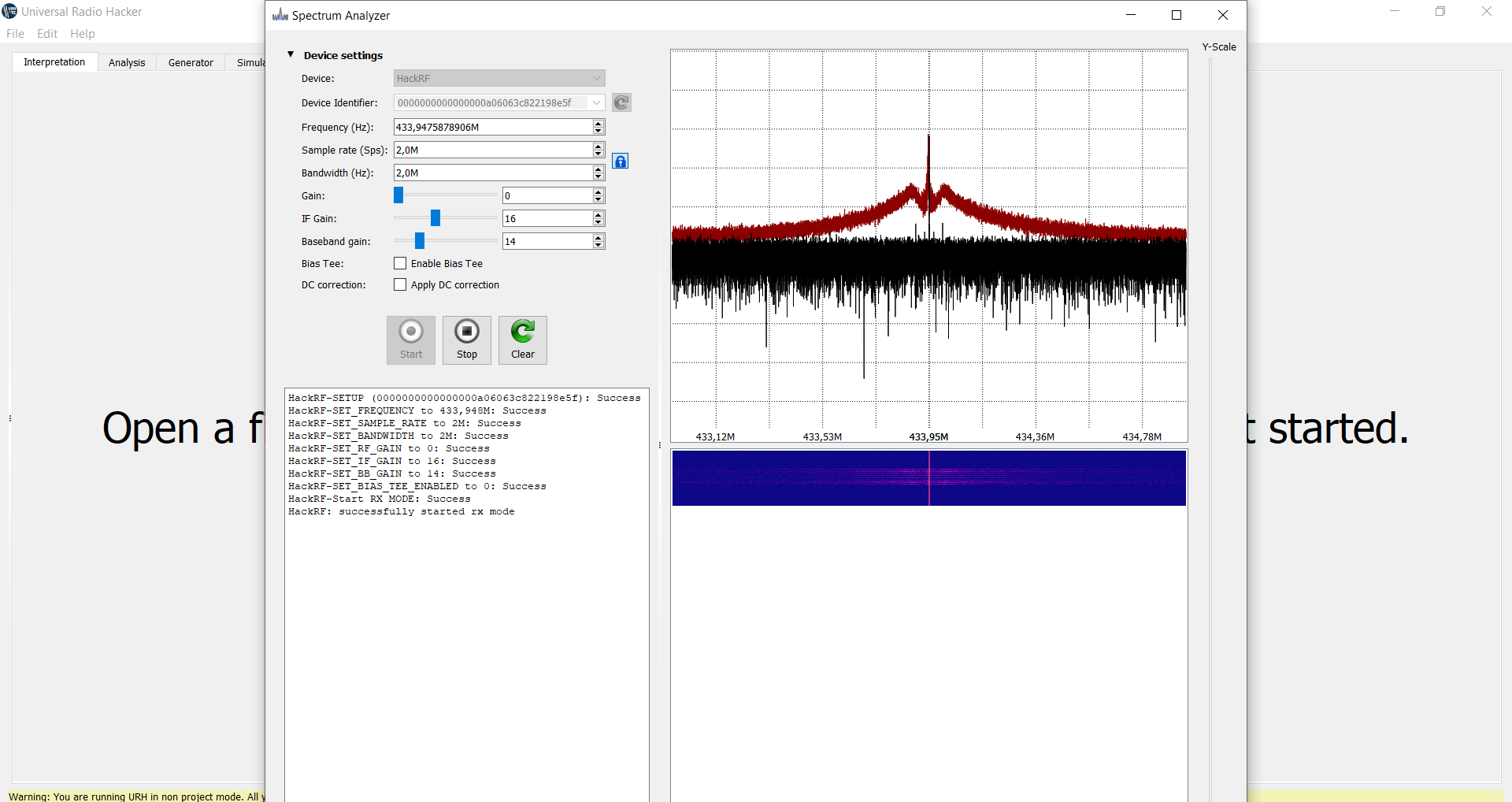
### HackRF replay napad

Korištenjem programa Universal Radio Hacker te HackRF One uređaja signal je ponovno snimljen te reproduciran i na taj način je izvršen „replay“ napad prikazan na slici 4.14.

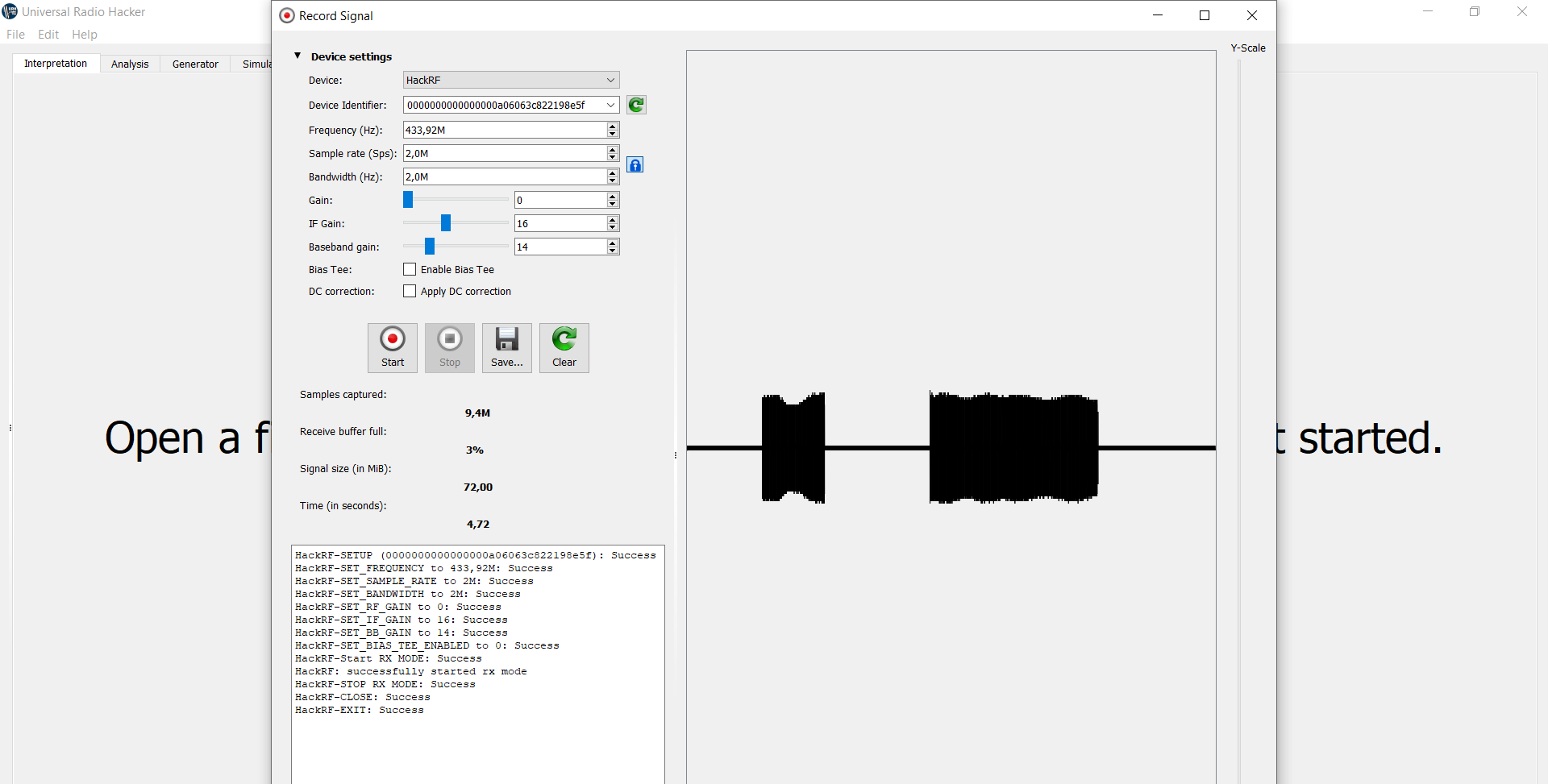


Slika 4.14 - HackRF replay napad

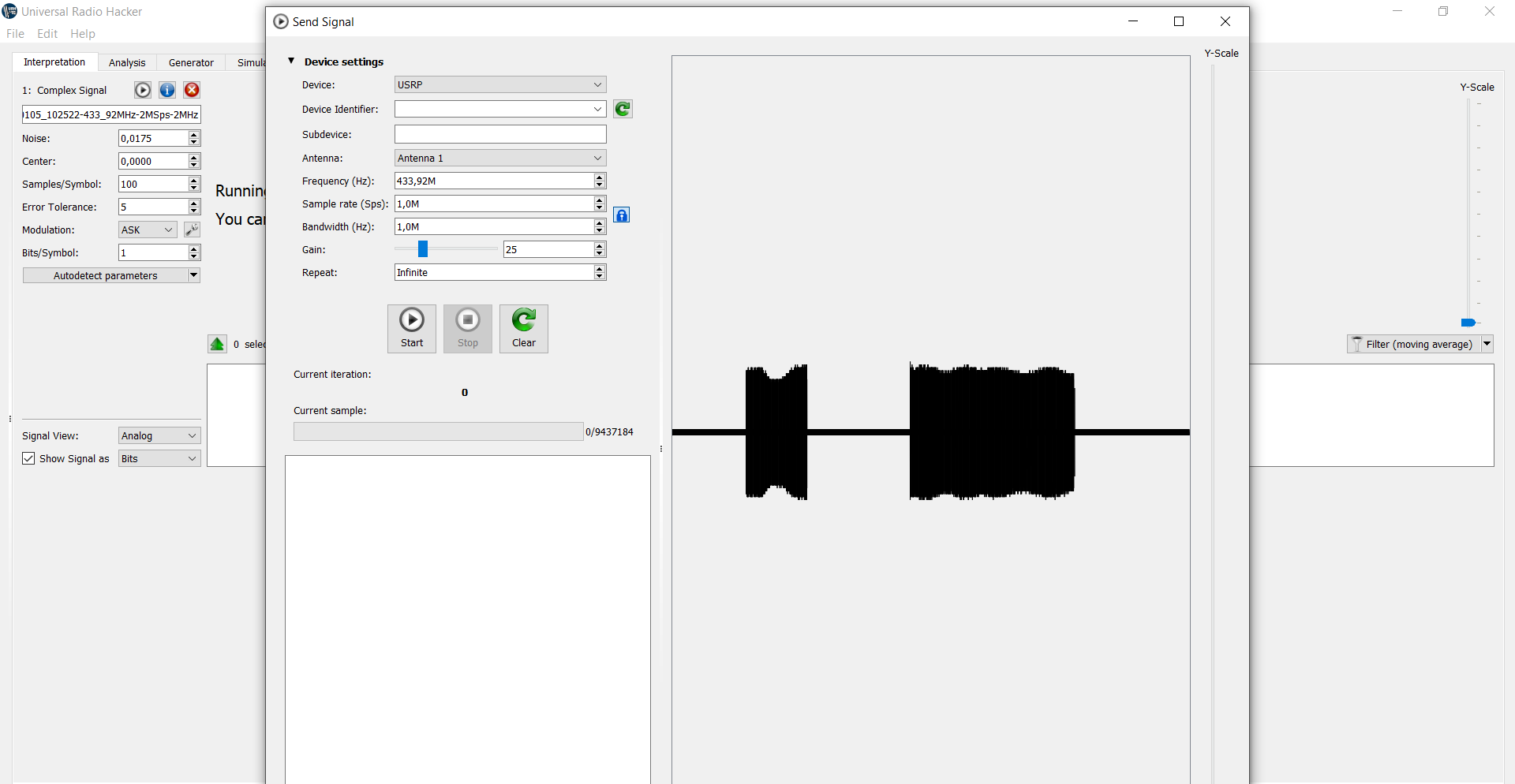
Zvono je zvonilo svaki put kad bi se reproducirao snimljeni signal što znači da je napad uspješno izvršen. Postupak je prikazan naslici 4.15(određivanje frekvencije na kojoj uređaj radi),slici 4.16(snimanje signala),slici 4.17te slici 4.18(reproduciranje signala, odnosno replay napad).



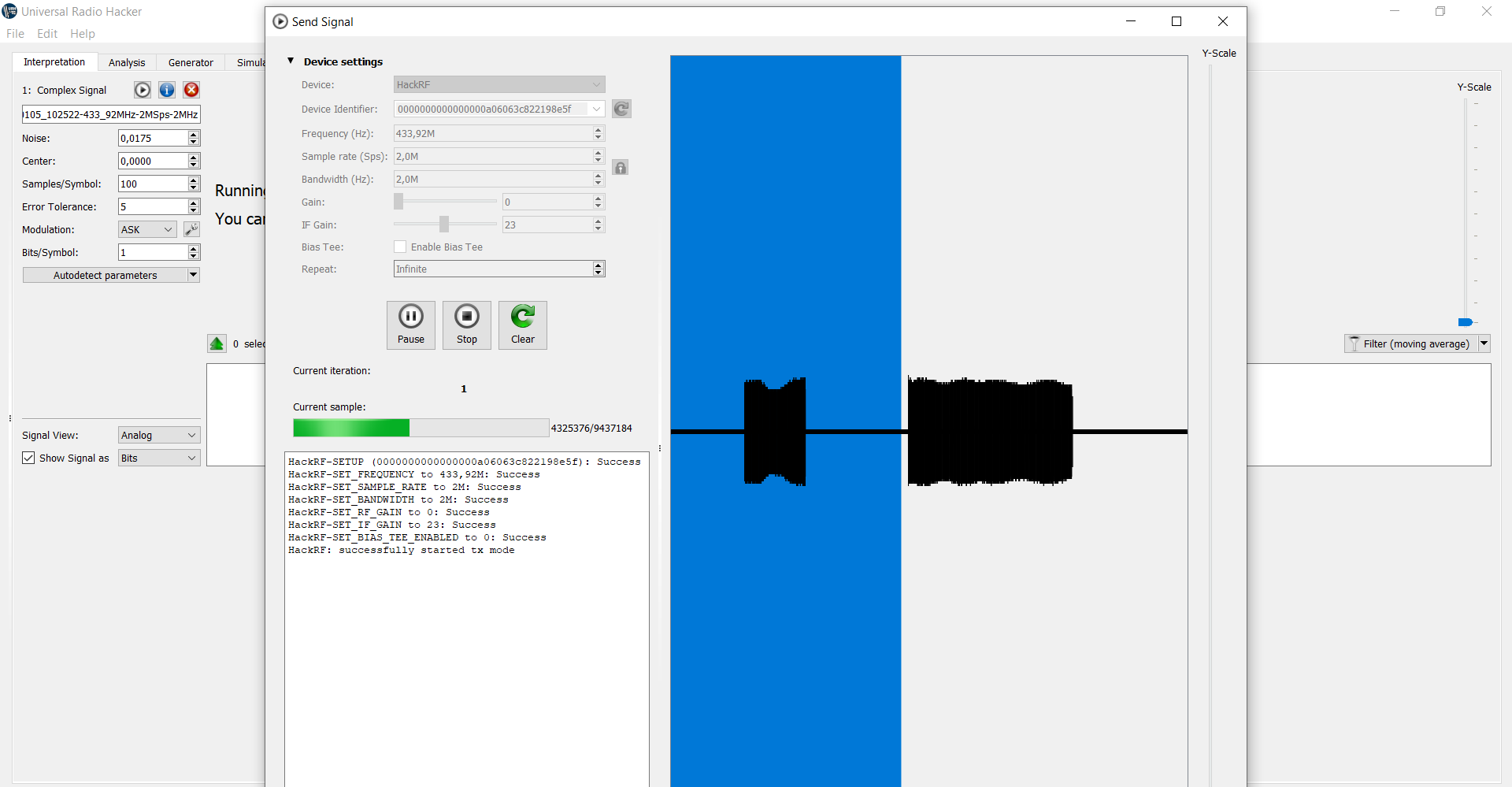
Slika 4.15 - Analiza spektra



Slika 4.16 - Snimanje signala



Slika 4.17 - Slanje signala



Slika 4.18 - Slanje signala

# Zaključak

Danas su domovi ispunjeni mnogim pametnim uređajima poput pametnih termostata, hladnjaka, televizora, utičnica, prekidača, zvona na vratima i mnogih drugih. Ti su uređaji često loše konfigurirani ili nesigurni po dizajnu, a prilično je lako hakirati ih zbog toga što ljudi ne paze čak ni na najosnovnije sigurnosne mjere.

U okviru ovog projekta pojašnjeno je hakiranje jednog od takvih uređaja, bežičnog zvona. Na početku je određena frekvencija na kojoj uređaj radi, a nakon toga je snimljen signal te analiziran u programu Audacity. Za određivanje frekvencije i snimanje korišten je program SDRSharp te HackRF uređaj. Na osnovu analize signala napisan je kod pokrenut na uređaju koji je napravljen spajanjem Arduina te predajnika. Napravljeni uređaj zazvonio bi bežično zvono svako 30 sekundi (to se vrijeme može proizvoljno mijenjati u kodu).

Osim toga, napravljen je i „replay napad“ na način da je signal snimljen i reproduciran u programu Universal Radio Hacker.

Na ovaj način može se hakirati i mnoge druge uređaje što dokazuje nesigurnost takvih uređaja.

# Literatura

1. <https://www.arduino.cc/en/Guide/ArduinoUno>
2. <https://store.arduino.cc/arduino-uno-rev3>
3. <https://greatscottgadgets.com/hackrf/one/>
4. <https://www.sparkfun.com/products/13001>
5. <https://randomnerdtutorials.com/rf-433mhz-transmitter-receiver-module-with-arduino/>

1. <https://www.chipoteka.hr/artikl/136579/prijemnik-i-predajnik-433-mhz-za-raspberry-i-arduino-mx-05vxd-rf-5v-8208000580> [↑](#footnote-ref-1)
2. <https://www.chipoteka.hr/artikl/117985/zvono-bezicno-52-melodije-bijelo-220v-emos-838w-8021370634> [↑](#footnote-ref-2)