Univerza v Ljubljani
Fakulteta za elektrotehniko

Projektna naloga
Digitalne komunikacije

UNIVERZALNI ODPIRAČ

Mentorji:
as. Timotej Gruden
doc. dr. Anton Umek

Amadej Vidic
Jakob Gazič
Mario Kjurchievski

April - Maj 2019

Kazalo vsebine

Ideja za projekt

Idejo za univerzalni odpirač smo dobili na laboratorijskih vajah, kjer smo se spoznali s pojmom SDR. Opazovali smo FM radijske postaje, nato pa smo ugotovili, da je mogoče namesto sprejemnika narediti tudi oddajnik. Takrat nas je zadeva začela vedno bolj in bolj zanimati.

Na fakulteti v LaIT smo si sposodili napravo HackRF, nato pa smo začeli iskati brezžične naprave, ki bi bile uporabne za naš projekt. Najprej smo začeli s preprostejšimi napravami.

Doma smo našli hišni zvonec, nato pa nam je z nekaj truda uspelo signal ponoviti tako, da je zvonec zazvonil tudi če nismo pritisnili na fizični gumb. Odločili smo se, da z raziskovanjem nadaljujemo in probamo analizirati še kakšno drugo (kompleksnejšo) napravo.

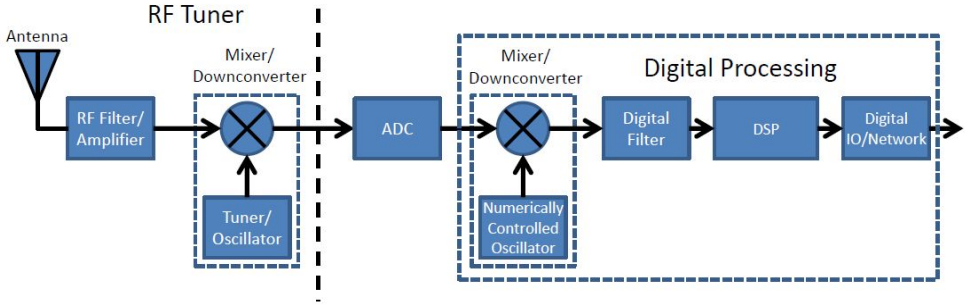
Z opazovanjem različnih signalov smo videli, da je za analiza ASK (AM) še enostavnejša od FM. Dobili smo idejo, da posnamemo signal za odklepanje, ki ga pošlje daljinec od avta. Uspelo nam je signal demodulirati in prebrati, kakšna koda je bila poslana. Iz demoduliranega signala pa je bilo lepo razvidno tudi, da je pri avtomobilskih ključih uporabljena rolling koda. Začeli smo razmišljati o tem, kateri oddajniki še uporabljajo AM in nimajo rolling kode. Bili smo presenečeni, koliko oddajnikov je takih in kaj vse bi se dalo nadzorovati in odpreti. Tako smo začeli s projektom Univerzalni odpirač.

Projekt smo 15. maja 2019 na Svetovni dan IKT tudi predstavili, ob 12. uri smo imeli v avli fakultete kratko predavanje.

Teoretični uvod v SDR

Uvod v Software – defined radio (SDR)

Software – defined radio (SDR) je radijsko - komunikacijski sistem, kjer se procesi, ki so se tradicionalno izvajali v strojni opremi (npr. filtri, ojačevalniki, modulatorji / demodulatorji, detektorji itd.), izvajajo s pomočjo programske opreme na osebnem računalniku.



Slika 1: SDR sprejemnik | Vir: https://www.curtisswrightds.com/content/images/Software-defined-radio.PNG

Nekaj lasnosti SDR-a:

- Relativno poceni in dostopni sprejemniki (RTL SDR)
- Možnost uporabe na zelo širokem frekvenčnem območju
- Glede (de)modulacije nismo strojno omejeni
- Zaradi velike skupnosti ni težko začeti

Zaradi svoje uporabnosti SDR zasledimo na številnih področjih, kot so:

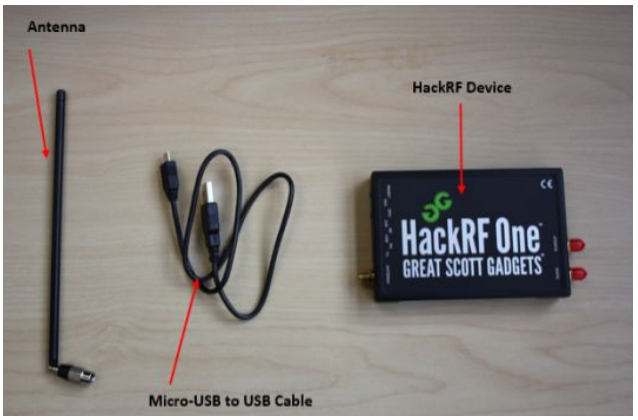
- *Mobilne komunikacije*: SDR je zelo uporaben na področjih, kot so mobilne komunikacije. Z nadgradnjo programske opreme se je mogoče prilagoditi na druge standarde brez potrebe po spremembah strojne opreme.
- *Raziskave in razvoj:* SDR je zelo uporaben v številnih raziskovalnih projektih. Radijske postaje je mogoče konfigurirati tako, da zagotavljajo natančne zahteve sprejemnika in oddajnika za vsako aplikacijo, brez potrebe po popolni konstrukciji strojne opreme.
- *Amaterska in domača uporaba*,
- *Vojska*

Slika 2: Različne SDR naprave | Vir: https://www.sdr-radio.com/Radios

HackRF One

HackRF One je SDR naprava, ki omogoča digitalizacijo radijskih signalov, ki jih lahko sprejema ali oddaja. Je ena izmed dražjih naprav (cca 300 €), njeni glavni prednosti pa sta možnost oddajanja ter zelo širok frekvenčni spekter (od 1 MHz do 6 GHz), kar vključuje veliko večino brezžičnih naprav. Naprava HackRF One deluje v poldupleks načinu, kar pomeni, da ne more sprejemati in oddajati hkrati.

Ob nakupu HackRF One dobimo samo napravo ter USB kabel, anteno pa moramo dokupiti. V našem primeru smo uporabili ANT 500 teleskopsko anteno, ki jo priporoča tudi proizvajalec Great Scott Gadgets.



Slika 3: HackRF One ter pripadajoča oprema | Vir: https://www.champlain.edu/Documents/LCDI/HackRF%20One%20Tutorial_F2017%20-%20Report.docx.pdf

Button/Light	Function
Reset Button	Used to reboot the HackRF One, equivalent to unplugging the device and plugging it back in ("HackRF One One," n.d.).
3v3 LED 1V8 LED RF LED	All three of these LEDs are used to indicate power and should be lit when the HackRF One is plugged in. The various colors are used to distinguish between the multiple LEDs on the side of the HackRF One ("FAQ," n.d.).
USB LED	Indicates that the HackRF One is communicating over USB ("FAQ," n.d.).
DFU Button	Used to install or update the firmware if it is not working properly or has never been installed ("HackRF One," n.d.).
RX LED	An orange light that indicates that the device is receiving information ("FAQ," n.d.).
TX LED	A red light that indicates that the device is transmitting information ("FAQ," n.d.).

Slika 4: Gumbi in lučke na HackRF One | Vir: https://www.champlain.edu/Documents/LCDI/HackRF%20One%20Tutorial_F2017%20-%20Report.docx.pdf

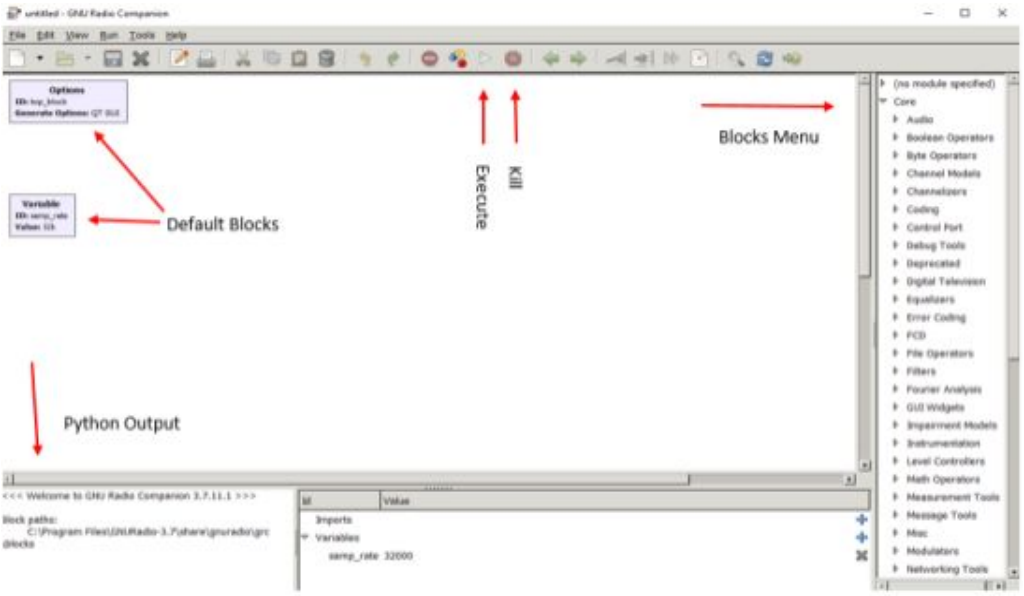
Software

GNU Radio Companion – je odprtokodni program, ki uporabnikom omogoča ustvarjanje grafičnega diagrama, s katerim sprejemamo, obdelujemo ter pošiljamo signale. Program ima vgrajenih veliko različnih modulov, kot so FM (de)modulacija, FFT, spektralni analizator,..

Za vhod ali izhod signala pa ima program podporo tudi za veliko SDR naprav (tudi HackRF).

Grafični diagram, ki ga naredimo se pretvori v Python kodo, računsko zahtevnejše operacije pa so sprogramirane v C++.

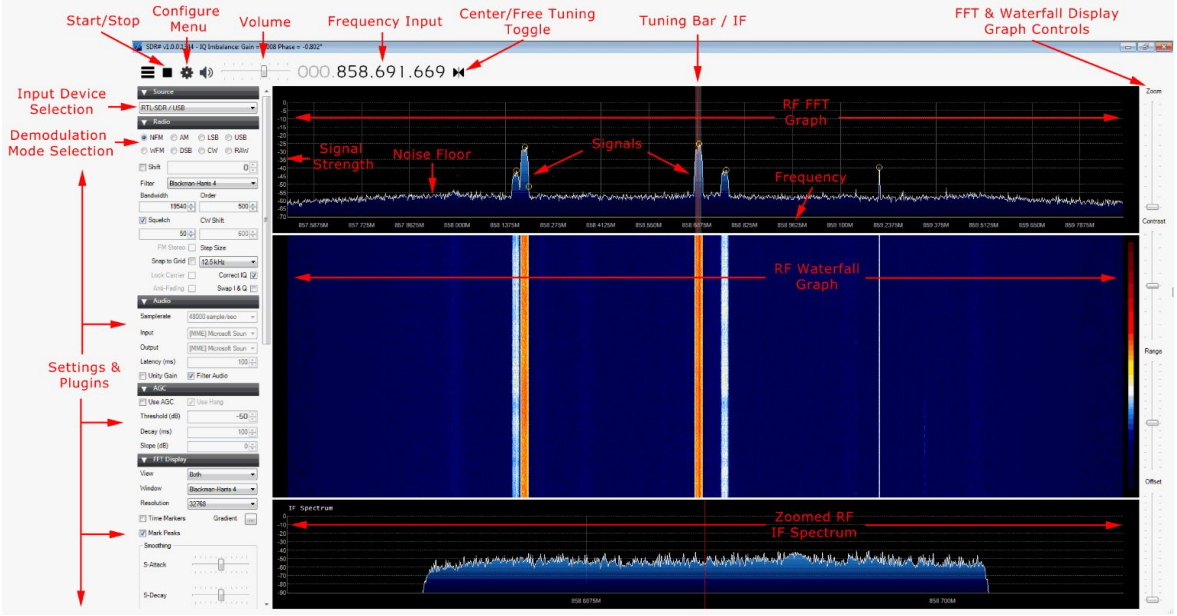
Program je napisan tako za Windows kot tudi za Linux sisteme.



Slika 5:GNU Radio Companion - začetno okno | Vir: https://www.champlain.edu/Documents/LCDI/HackRF%20One%20Tutorial_F2017%20-%20Report.docx.pdf

SDR# - je eden izmed najpreprostejših SDR programom za uporabo. Najprimernejši je za opazovanje frekvenčnega spektra ali za poslušanje (demodulacijo) analognih zvočnih signalov (med drugim AM, NFM, WBFM, CB, CW). Pri projektu smo ga uporabili za iskanje frekvence, na kateri deluje neka neznana naprava.

Program je napisan le za Windows sisteme.



Slika 6: SDR# z oznakami | Vir: https://www.rtl-sdr.com/sdrsharp-users-guide/

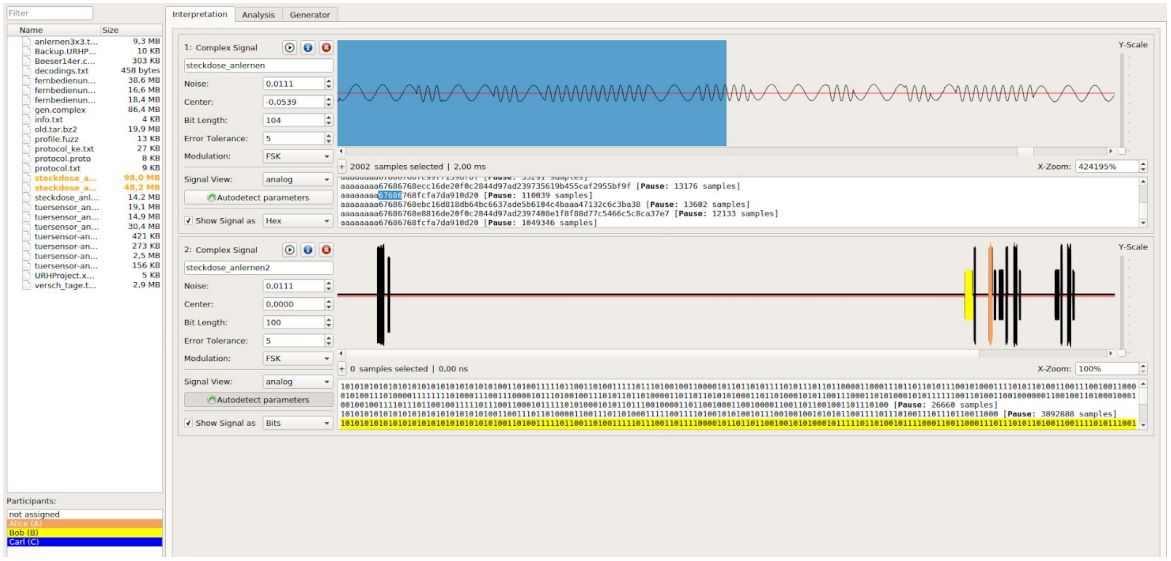
Universal Radio Hacker (URH) - je odprtokodna programska oprema za raziskovanje in analizo (neznanih) signalov ter protokolov. Uporablja se predvsem za demodulacijo in dekodiranje signalov. URH razdeli postopek analize na več faz:

- Interpretacija
- Analiza
- Generiranje
- Simulacija

  pri čemer se rezultati iz ene faze lahko prenesejo na drugo. Programska oprema ponuja vse funkcije, ki so potrebne za proučevanje protokola in za uporabnika niso preveč zapletene.

Med drugimi uspešno demodulira tudi ASK, FSK in PSK.

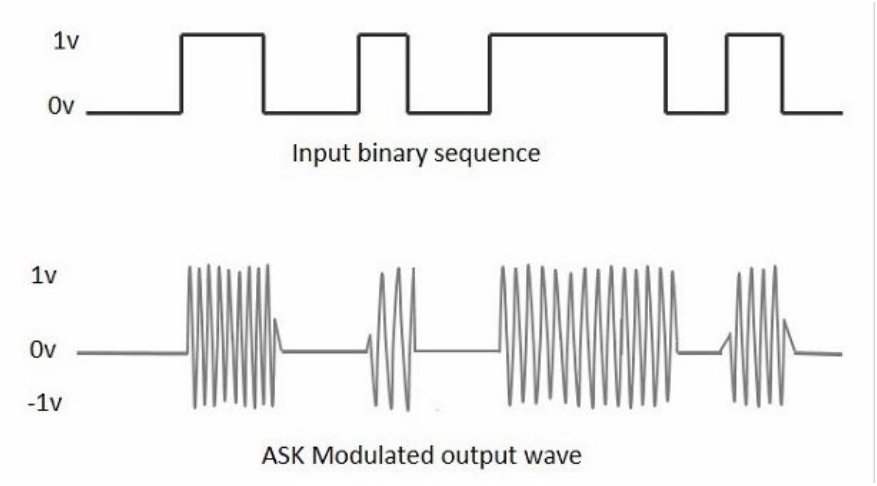
Program je napisan tako za Windows kot tudi za Linux sisteme.



Slika 7: Universal Radio Hacker - primer opazovanega signala | Lastni vir

Amplitudna modulacija

Amplitudna modulacija (kratica AM) je način prenašanja informacij na daljavo s spreminjanjem jakosti oz. amplitude nosilnega signala s konstantno frekvenco. Najpogosteje se uporablja za komunikacijo z radijskim valovanjem in je obenem najstarejši način za radijski prenos zvoka.
ASK je vrsta amplitudne modulacije za oddajanje digitalnega signala - kode. Vidimo, da je pošiljanje kode precej enostavno, za 1 imamo signal neke frekvence z fiksno amplitudo, za 0 pa signala nimamo.



Slika 8: ASK moduliran signal | Vir:https://www.tutorialspoint.com/digital_communication/digital_communication_amplitude_shift_keying.htm

DEMODULACIJA: Pri demodulaciji vzamemo absolutno vrednost in jo damo na detektor ovojnice. Na sliki je lepo razvidno, da iz prebrane ovojnice dobimo ravno poslan signal. Če želimo vedeti še točno kodo, ki je bila poslana moramo ugotoviti le še dolžino enega bita. Ko imamo vse to lahko zapišemo točno zaporedje bitov poslanega signala.

Prednosti in slabosti amplitudne modulacije:
Prednosti:

- Glavna prednost je enostavnost (tako oddajnik, kot sprejemnik sta zelo enostavna)
- Cena, sprejemnik in oddajnik sta enostavna in zato razmeroma poceni

Slabosti:

- Sprejem AM signala je močno občutljiv na naravne in umetno povzročene motnje iz okolja

ISM band, frekvence

(**I**ndustrial**, s**cientific and **m**edical **(ISM)** radio bands)

Tako imenujemo dele radijskega spektra, ki so namenjeni za industrijsko, izobraževalno in medicinsko rabo. Glavna lastnost, ki velja za te pasove je, da je na njih pod določenimi pogoji dovoljeno oddajati vsakemu posamezniku. Pri našem projektu smo opazili, da je najbolj popularen 433 MHz frekvenčni pas (poleg 2.4 GHz).

Frequency range		Center frequency	Bandwidth	Type
6.765 MHz	6.795 MHz	6.78 MHz	30 kHz	A
13.553 MHz	13.567 MHz	13.56 MHz	14 kHz	B
26.957 MHz	27.283 MHz	27.12 MHz	326 kHz	B
40.66 MHz	40.7 MHz	40.68 MHz	40 kHz	B
433.05 MHz	434.79 MHz	433.92 MHz	1.74 MHz	A
902 MHz	928 MHz	915 MHz	26 MHz	B
2.4 GHz	2.5 GHz	2.45 GHz	100 MHz	B
5.725 GHz	5.875 GHz	5.8 GHz	150 MHz	B

Slika 9: ISM band (frekvence, kjer lahko oddajamo) | Vir:https://en.wikipedia.org/wiki/ISM_band

Eden izmed izjem, ki ni uporabljal ISM pasu pa je bil eden od daljincev za garažo, ki deluje na frekvenci 868 MHz.

Slika 10: Daljinec za garažo | Lastni vir

Projekti

Ko smo se enkrat spoznali z osnovami HackRFa, SDR programsko opremo, frekvenčnimi pasovi ter modulacijami smo se lotili prvih projektov.

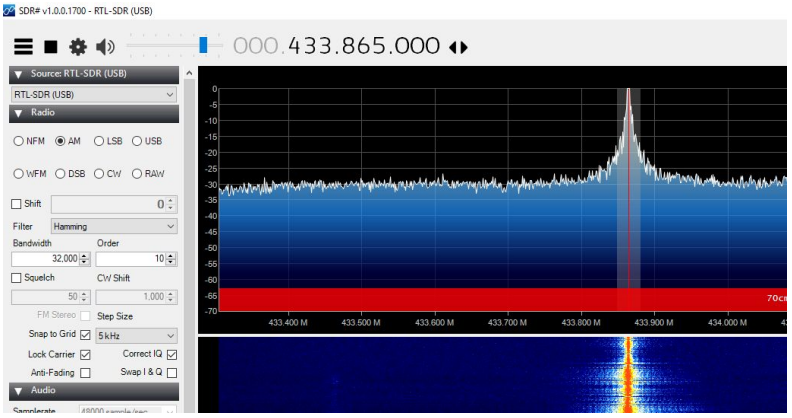
Hišni zvonec

Prvi projekt, ki smo se ga lotili je hišni zvonec. Za to smo se odločili, ker se nam je zdelo, da ta naprava ne bo uporabljala kakšnih posebnih protokolov ali algoritmov - torej, da bo z njim lažje začeti.
Hišni zvonec je naprava, ki jo vstavimo v vtičnico v notranjem prostoru hiše, pozvonimo pa ga z daljincem, ki deluje na baterijsko napajanje.

Slika 11: Brezžični zvonec | Lastni vir

Določi frekvenco

Prva naloga, ki nas je čakala je bila, da ugotovimo, na kateri frekvenci naš zvonec deluje. Žal nismo imeli te sreče, da bi bila frekvenca zapisana na hrbtnem delu zvonca ali daljinca, tako da smo jo morali poiskati sami. Naloga se je na začetku zdela skoraj nemogoča, vendar smo po raziskovanju ugotovili, da mora vsaka taka naprava delovati na ISM bandu. Po hitri analizi frekvenčnega spektra s programom SDR# smo frekvenco našli.
Frekvenca oddajnega signala se je nahajala na pogosto uporabljenem 433 Mhz območju, oziroma malce natačneje, frekvenca je bila **433.883 Mhz**.

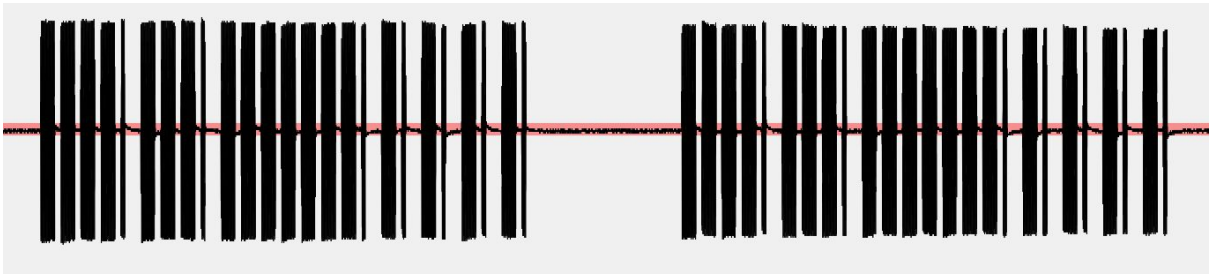


Slika 12: Signal zvonca v SDR # | Lastni vir

Demodulacija signala in koda

Po določitvi frekvence je bil naslednji korak, da signal analiziramo.
Signal smo najprej posneli z programom Universal Radio Hacker (v nadaljevanju URH). Po vizualnem pregledu nam je bilo hitro jasno, da imamo opravka z ASK (Amplitude Shift Keying) modulacijo.

Slika 13: Celoten signal zvonca v URH | Lastni vir



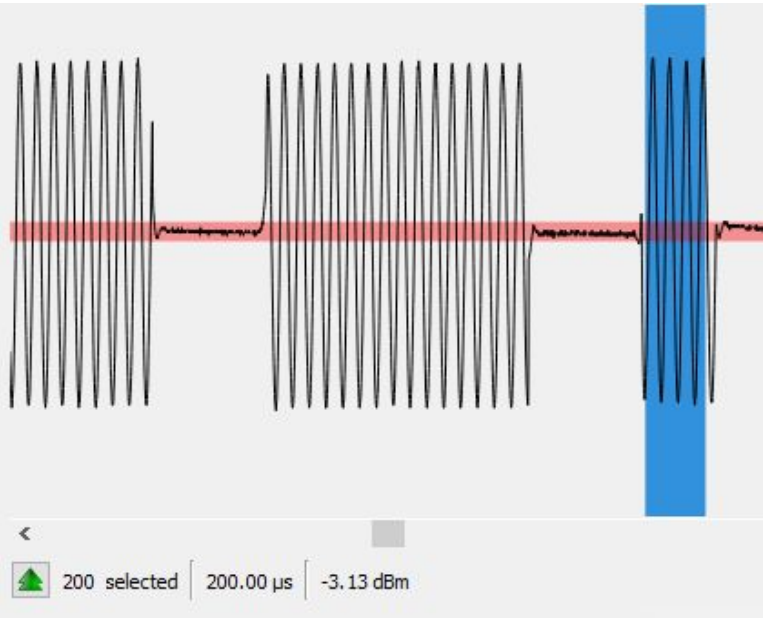
Slika 14: Približan signal zvonca v URH | Lastni vir

Ugotavljanje parametrov: da lahko iz ASK moduliranega signala izluščimo poslano kodo moramo poznati 3 glavne lastnosti takega signala in modulacije, to so:

- noise ratio
- dolžina bita (časovno)
- koda

Noise ratio
Da lahko razberemo informacijo, moramo določiti neko vrednost, ki je nad nivojem šuma in pod najnižjim nivojem signala, da lahko program določi kodo. Vrednost najlažje določimo vizualno, v našem primeru je znašala **0.0278.**

Dolžina bita
Da iz signala lahko dobimo informacijo, ki je zapisana z biti, moramo vedeti, kolikšna je sploh dolžina enega bita. Dolžino bita merimo v številu vzorcov na sekundo, oziroma jih glede na frekvenco vzorčenja preračunamo v mikrosekunde. V našem primeru je znašala **200 mikrosekund**.
Tudi tu dolžino bita določimo vizualno. To storimo tako, da v signalu najdemo najkrajši del (enke ali ničle), nato pa ga izmerimo. Pri tem načinu seveda obstaja verjetnost da bi naredili napako, če se na primer v kodi en sam zaporedni (isti) bit sploh ne bi pojavil (bi se pa na primer pojavili 3 zaporedni). Kljub temu je ta metoda dosti dobra, saj je pri dolžini kode, ki se pošilja verjetnost za kaj takega zelo majhna. Če bi do napake prišlo pa bi si signal še enkrat podrobneje ogledali.



Slika 15: Dolžina enega bita (1) v URH | Lastni vir

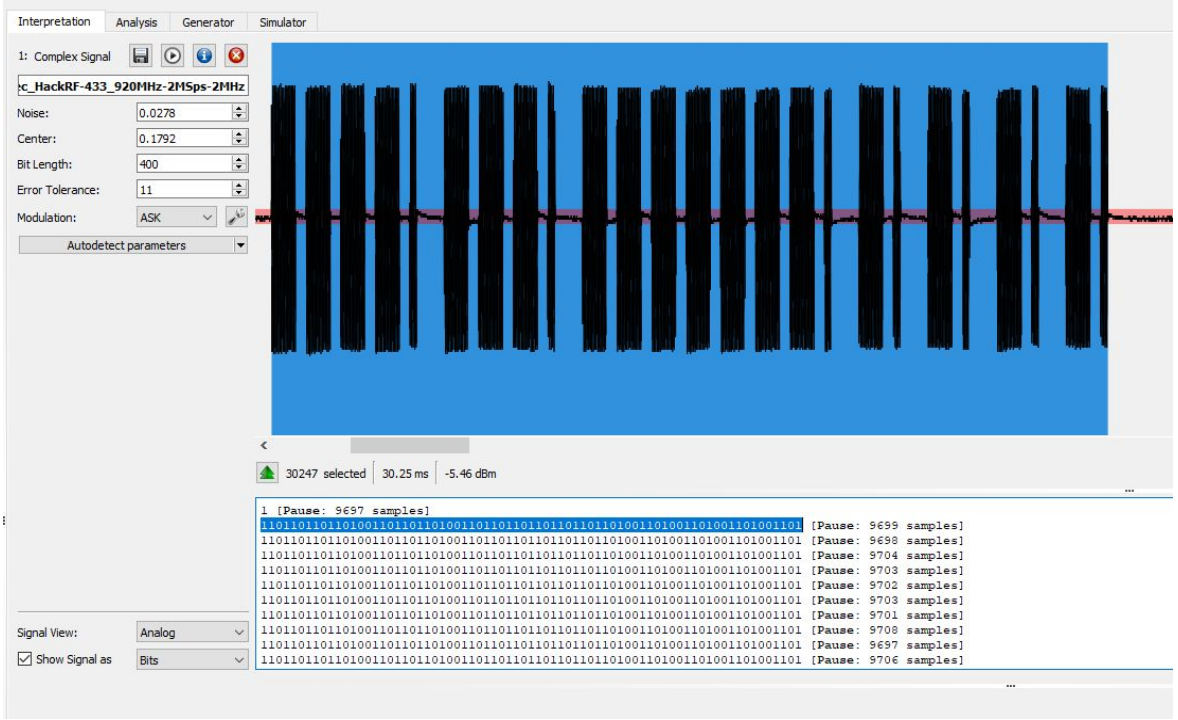
Koda - string bitov

V kolikor smo zgornje dva parametra pravilno določil, nam bo URH samodejno prikazal kodo. Koda je zapisana v bitih, za lažji pogled pa program omogoča, da si jo ogledamo tudi v šestnajstiškem (hex) zapisu.

Koda v bitih:

“**1101101101101001101101101001101101101101101101101001101001101001101001101**
”

Koda v hex: “**db69b69b6db69a69a68”**



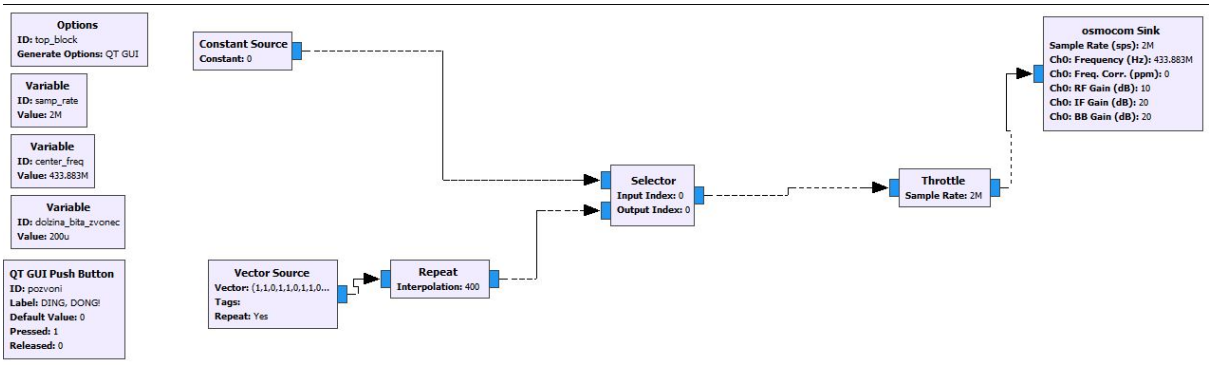
Slika 16: Koda v bitih (URH) | Lastni vir

Interpretacija kode

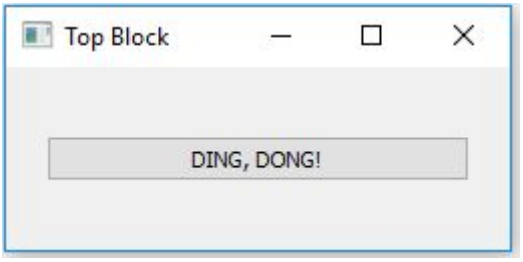
Ko enkrat pritisnemo na gumb, se po zraku pošlje signal, ki vsebuje zgornjo kodo. Opazimo, da dalj časa kot gumb držimo, večkrat se ta ista koda pošlje oziroma ponovi. Predvidevamo, da gre za mehanizem, ki napravo naredi bolj robustno (če na primer sprejemnik prvič zaradi neke motnje kode ne sprejme pravilno, jo bo naslednjič). Glavna ugotovitev je torej, da vsakič ko pritisnemo gumb, daljinec zvoncu pošlje **isto kodo**. V tem trenutku smo torej na konju in lahko nadaljujemo z izdelavo našega lastnega oddajnika.

Izdelava oddajnika - daljinca

Ker se nam je ideja o izdelavi našega lastnega daljinca zdela zanima, smo se odločili, da se izdelave lotimo.
Za to nalogo smo uporabili program GNU Radio, shema našega programa pa je razvidna na spodnji sliki.



Slika 17: Shema diagrama daljinca v GNU Radio | Lastni vir



Slika 18: Grafičen gumb, s katerim pozvonimo | Lastni vir

Opis delovanja GNU Radio programa

Spremenljivke, ki smo jih iz signala izluščili s pomočjo URH shranimo v program. Samo bitno kodo shranimo kot vektor (Vector Source). Nato nastavimo še Selector, to je blok, ki izbira ali se bo naša koda pošiljala ali ne. V našem primeru je to odvisno od tega, če držimo gumb “DING, DONG!” ali ne. Naslednji blok je Throttle. Njegov namen je, da se naš vektor ne bi bral ter pošiljal prehitro, ampak z pravo hitrostjo. Na koncu pa signal peljemo še na blok Osmocom Sink, ki ga nastavimo na pravo frekvenco. Ta končni blok je povezava na našo fizično napravo HackRF - signal se odda.

Po odpravi nekaj tehničnih težav nam je program uspelo usposobiti - razvili smo svoj daljinec, ki deluje odlično.

Ladjica na daljinsko vodenje

Po uspešno opravljenem projektu Hišni zvonec, smo se lotili ladjice na daljinsko upravljanje. Ker je bil sistem poteka zelo podoben tistemu pri zvoncu, bomo tu izpostavili le ključne razlike in zanimivosti.

Slika 19: Ladjica na daljinsko upravljanje | Lastni vir

Frekvenca

Za razliko od zvonca je bil tu proizvajalec tako prijazen, da nam je frekvenčno območje napisal kar na prednjo stran daljinca - 27 MHz.
S pregledom SDR# smo frekvenčno območje potrdili, točna frekvenca, ki smo jo izmerili pa je **27.145 Mhz**.
Glede na sliko 9 vidimo, da je ta frekvenca del ISM frekvenčnega pasu.

Demodulacija in koda

Tudi tu hitro opazimo, da imamo opravka z ASK modulacijo. Princip je enak kot pri zvoncu, z razliko, da ima tukaj vsak ukaz svojo kodo. Vseh možnih ukazov je 8, za demonstracijo pa smo demodulirali le glavne 4 (naprej, nazaj, levo, desno).

Dožina bita: 500 mikrosekund

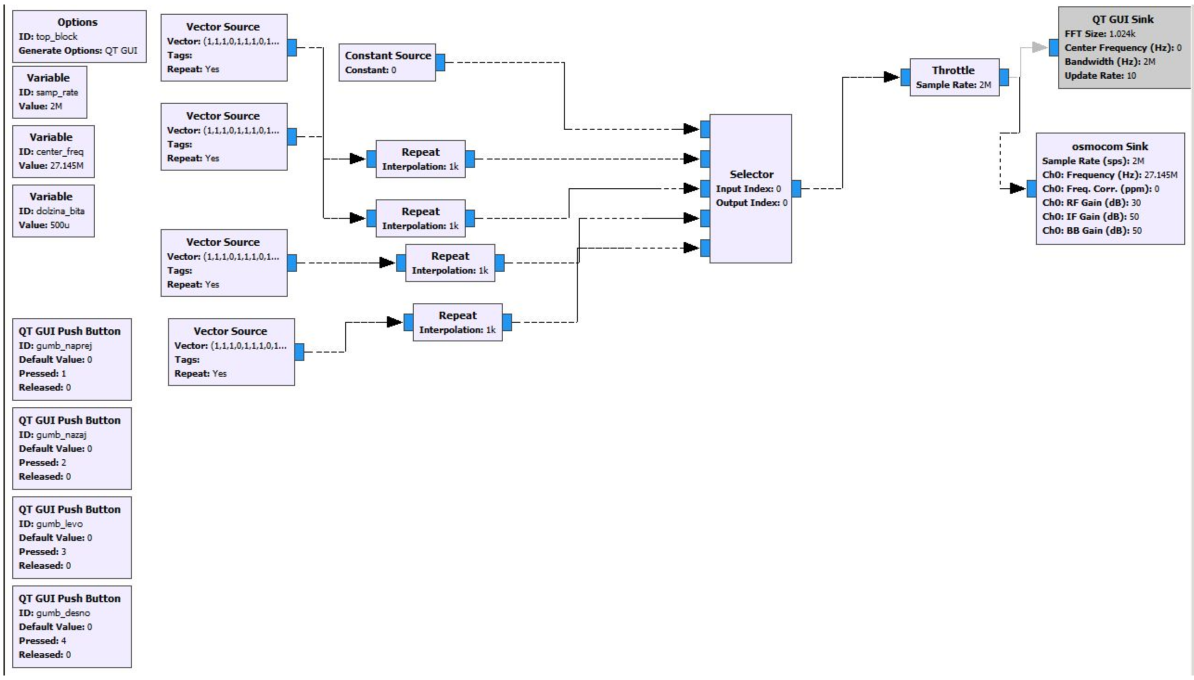
Koda_naprej:
(1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0)
Koda_nazaj:
(1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0)

Koda_levo:
(1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0)

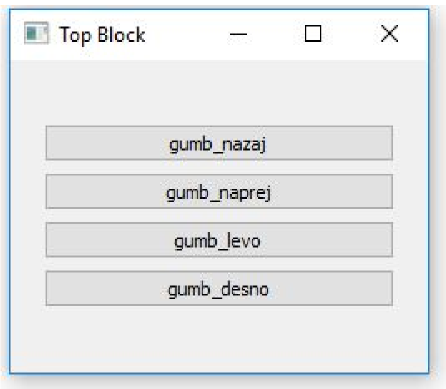
Koda_desno:
(1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
,1,0,1,0,1,0,1,0,1,0,1,0)

Izdelava oddajnika

Program je zelo podoben tistemu pri zvoncu, glavna razlika je, da ga tu razširimo na 4 različne gumbe oziroma signale.



Slika 20: Shema programa oddajnika ladjice - GNU Radio | Lastni vir



Slika 21: Gumbi za krmiljenje ladjice - GNU Radio | Lastni vir

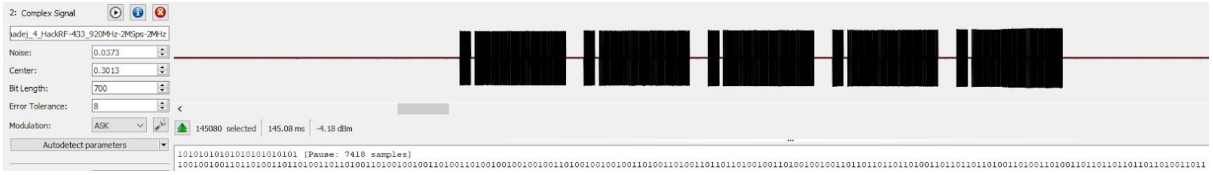
Garaže

Čeprav se nam je zdela verjetnost, da pomembnejše stvari kot so garaže ne bi imele implementiranega nekega varnostnega protokola majhna, smo se odločili, da vseeno preverimo, kaj se dogaja. Ker je postopek demodulacije enak kot pri prejšnjih dveh projektih (tudi garaže uporabljajo ASK) se bomo tu osredotočili bolj na kodo oziroma protokol.
Za testiranje smo imeli na voljo 2 garaži.

Garaža #1

Prva garaža deluje na frekvenci 433 MHz, kar je razvidno tudi iz hrbtne strani daljinca.

Po pregledu signala z URH ugotovimo, da je koda dolga 196 bitov.



Slika 22: Signal odpiranje/zapiranje garaže #1 (v URH) | Lastni vir

Opazili smo, da se ob vsakem pritisku gumba na daljincu pošlje druga koda, sprejemnik v garaži pa se ne odzove, če mu pošljemo staro, kar smo tudi preizkusili. Na hiter pogled tega nismo videli, saj sta si kodi v prvi polovici enaki. Predvidevamo, da je prva polovica ID garaže, druga pa geslo.
Po raziskovanju smo ugotovili, da garaža uporablja tako imenovan Rolling Code sistem. Sistem deluje na preprost način: oddajnik ima zapisano neko funkcijo, ki za vsak naslednji pritisk daljinca generira novo kodo. Tudi oddajnik ima zapisano to isto funkcijo, zato ve, katero kodo mora pričakovati. Sistem ima sicer nekaj pomanjkljivosti, vendar je v splošnem dobra zaščita pred ljudmi, ki nimajo odobrenega vstopa.

Garaža #2

Tudi pri drugi garaži frekvenco preberemo iz daljinca. V tem primeru gre za frekvenčno območje 868 MHz.

Po pregledu signala z URH ugotovimo, da je koda dolga 97 bitov. Na naslednji sliki smo kodo namenoma izpustili.

Slika 23: Signal odpiranje/zapiranje garaže #2 (v URH) | Lastni vir

Na naše veliko presenečenje opazimo, da je poslana koda ob vsakem pritisku gumba enaka in se ne spreminja. Sprejemnik ter oddajnik torej nimata implementirane čisto nobene

varnosti, kar pomeni, da odpreti tako garažo ni nič težje, kot je na primer pozvoniti na zvonec iz prvega projekta! Kakšne so možnosti zlorabe si lahko predstavljate sami, povemo pa lahko še, da je zraven te garaže še veliko garaž istega tipa.

Slika 24: Odpiranje garaže z Univerzalnim odpiračem | Lastni vir

IDEJE ZA NADALJEVANJE

Ker nam je je bil projekt zelo zanimiv se bomo s to tematiko verjetno v prihodnje še ukvarjali. Za nadaljevanje imamo že nekaj zanimivih idej.

Začeli bi z napravami, ki uporabljajo malce naprednejšo komunikacijo.
Videli smo že, da lahko prestrežemo signal TPMS (tyre pressure monitoring system), ki se uporablja pri novejših avtomobilih. Če bi se uspeli naučiti protokola, ki ga sistem uporablja, bi lahko avtu poslali drugačne podatke in opazovali kaj bi se zgodilo. Podoben princip bi lahko prenesli tudi na druge naprave, na primer, lahko bi simulirali senzor za dim, sprejemniku bi lahko pošiljali poljubne vrednosti (enako velja za brezžične vremenske postaje).

Še en zelo zanimiv primer so brezžične tipkovnice. Zanima nas, če bi se iz prestreženega signala dalo razbrati, kaj je uporabnik napisal na tipkovnico. Zasledili smo namreč, da uporabljena enkripcija pri nekaterih modelih ni najboljša.

Naslednja zelo zanimiva zadeva, ki bi se je lotili pa je rolling code. Ta zaščita je sicer razmeroma dobra in varuje veliko zadev, vendar obstaja način, kako tako zaščito lahko zaobidemo. Projekta bi se lotili tako, da bi ujeli signal, ki ga sprejemnik (npr. avtomobilski) še ni slišal, potem pa bi ta signal uporabili kasneje. To bi storili tako, da bi z enim oddajnikom pošiljali šum in tako zmotili signal, ki ga je poslal daljinec. S svojim SDR sprejemnikom bi nato signal prestregli in šum odstranili. Vse to bi seveda testirali na lastnih avtomobilih.