SPARKDEV

ΥΠΕ

CyberSecurity

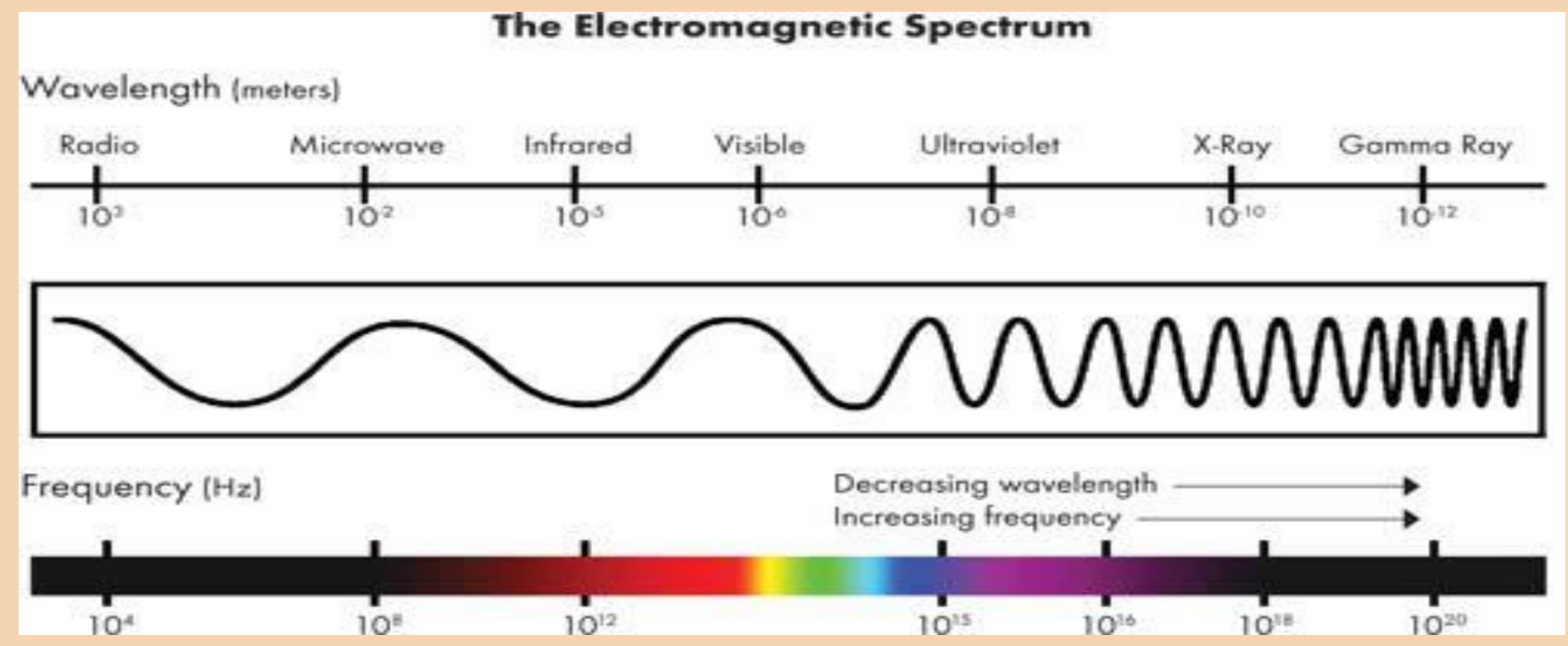
Shaneka Lewis & Morgan Draine

Franklin Abreu, Carlos Marquez, Yaohua Hu, Jamsher Nigmatulloev, Jose Taleno, Aaron Turransky, Levi Le, Khrisanni Brown, Michael Vigil, Kyle Leibovitz, Veronica Canido, Nathaly Hernandez

Introduction

In the electromagnetic spectrum, the radio frequency (RF) band (20 kHz - 300 GHz) is used as a form of wireless communication. Standards for RF communication include AM, FM, Bluetooth and WiFi. The growing trend of "smart" devices are devices that are using RF to communicate, interact, and share data with other devices. These devices that use RF standards include but are not limited to phones, computers, IoT devices and remotes.

Cybersecurity has placed a focus on securing wireless communication in response to the exponential growth of connected devices per person. 2018 was the first year that the number of connected devices outnumbered the number of people on Earth, and is only continuing to grow. By its definition, the smart devices you possess such as your phone, watch, and home appliances have some sort of wireless communication. While increased communication has many benefits, it also means that there are more opportunities for vulnerabilities. With all this communication between devices how secure are your things? How easy is it to "hack" something as important as a key fob to your car?



Of the many attack types that are possible in the cybersecurity field, our team has decided that the best approach to exploiting RF communication is through a replay attack. A replay or relay attack is a method in which seemingly innocent information is captured by intercepting a legitimate transmission, storing the information and using it later for the attacker's gain.

Objective

1. Develop a device that will:
   a. Capture and save information from rf devices operating at 315 MHz or 433 Mhz
   b. Transmit that information at a time of our choosing to exploit some system
2. Create test environments to evaluate our device
3. Test device using test environments:
   a. Scenario 1: RF transmitter sending text to a receiver
   b. Scenario 2: "Garage Door" Key fob turning on LEDs
4. Copy data transmitted from Key Fob of a car and open the car using our device

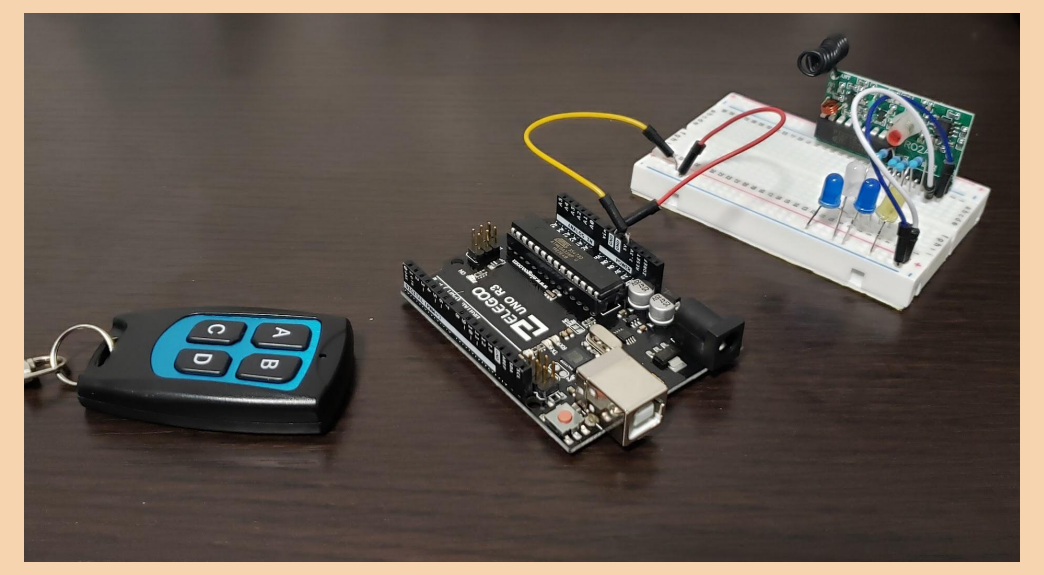
Materials

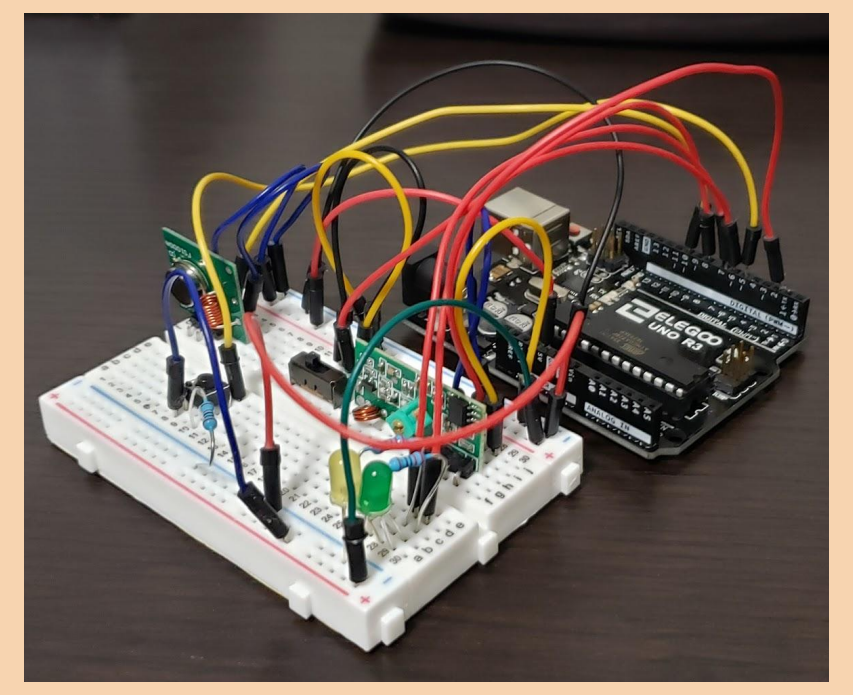
Hardware:

- Arduino Uno
- 315 MHz Receiver and Transmitter
- 433 Mhz Receiver and Transmitter
- Nooelec R820T SDR & DVB Tuner
- Adafruit Key Fob and Momentary receiver

Software:

- Arduino IDE
- SDRSharp
- Audacity
- RTL_433 library

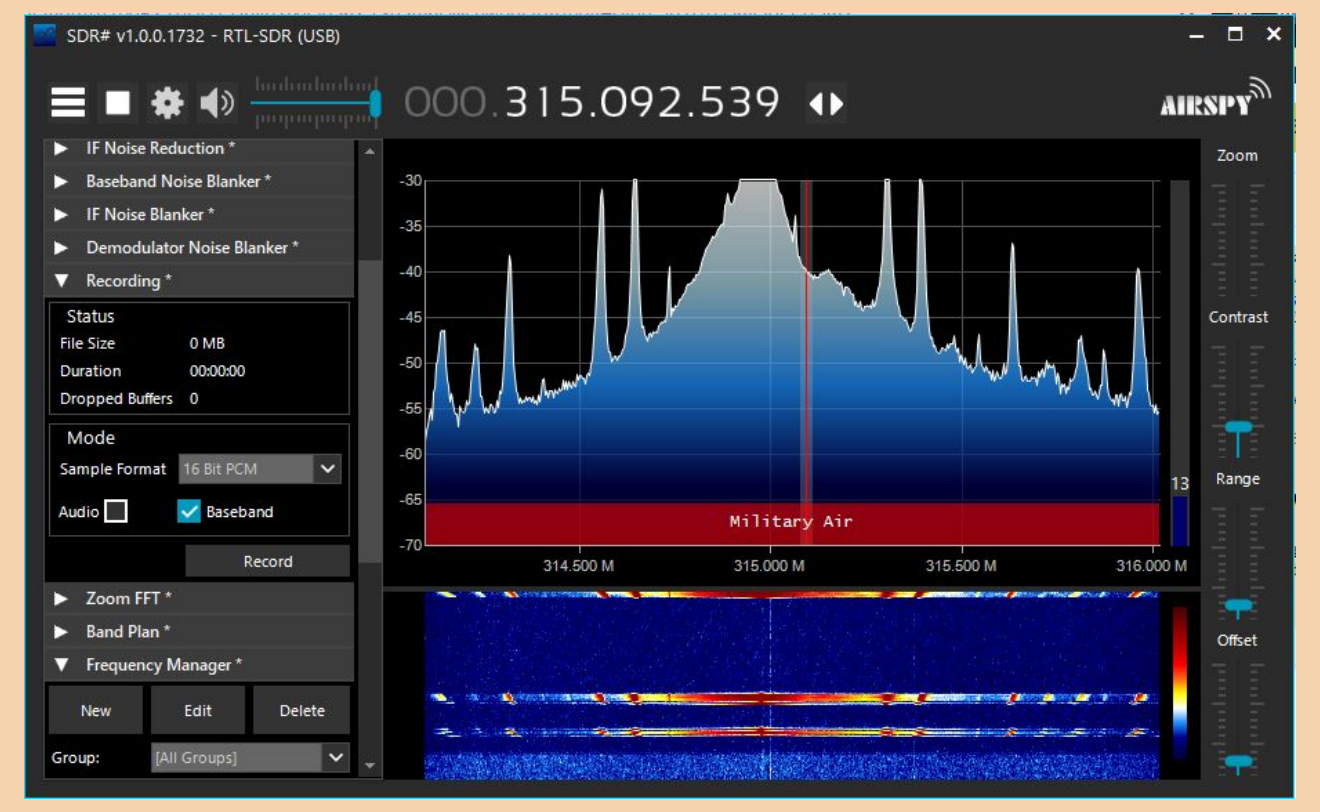




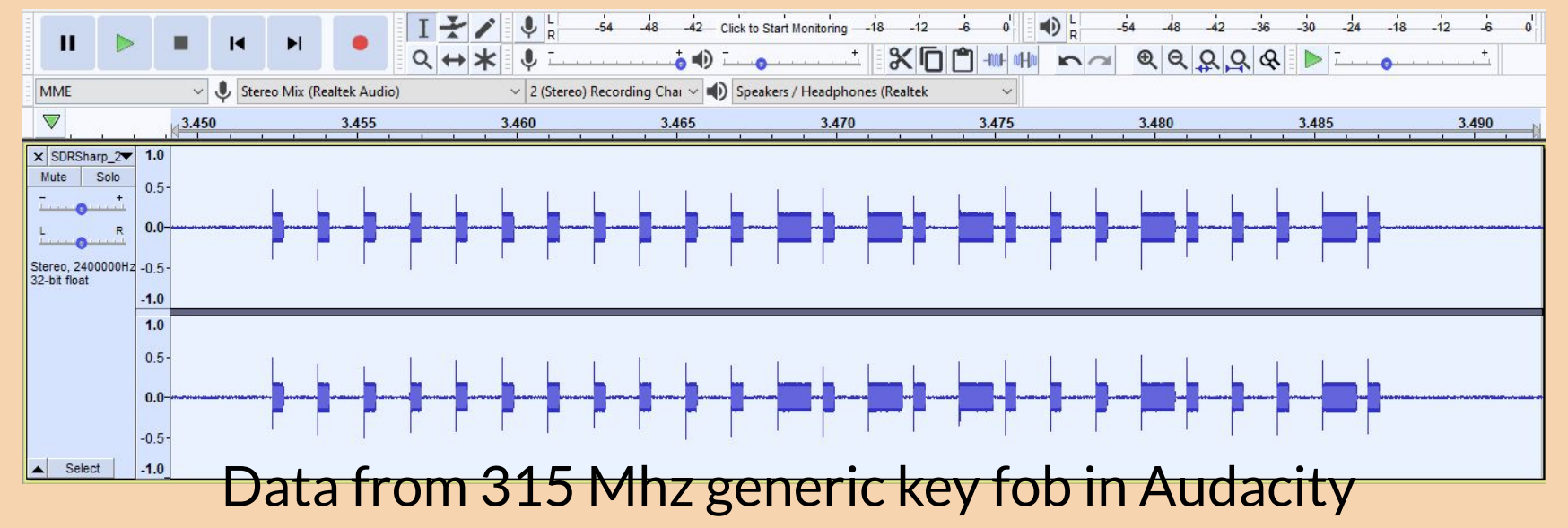
Conclusions

- We were successfully able to intercept and transmit traffic from both test environments using our device.
- Companies that specialize in the development of devices that communicate using radio frequency have implemented protocols to mitigate relay attacks although these are not always perfect
  - Rolling code / hopping codes
  - RF packets are longer with more identifiers
  - RF isolation pouches

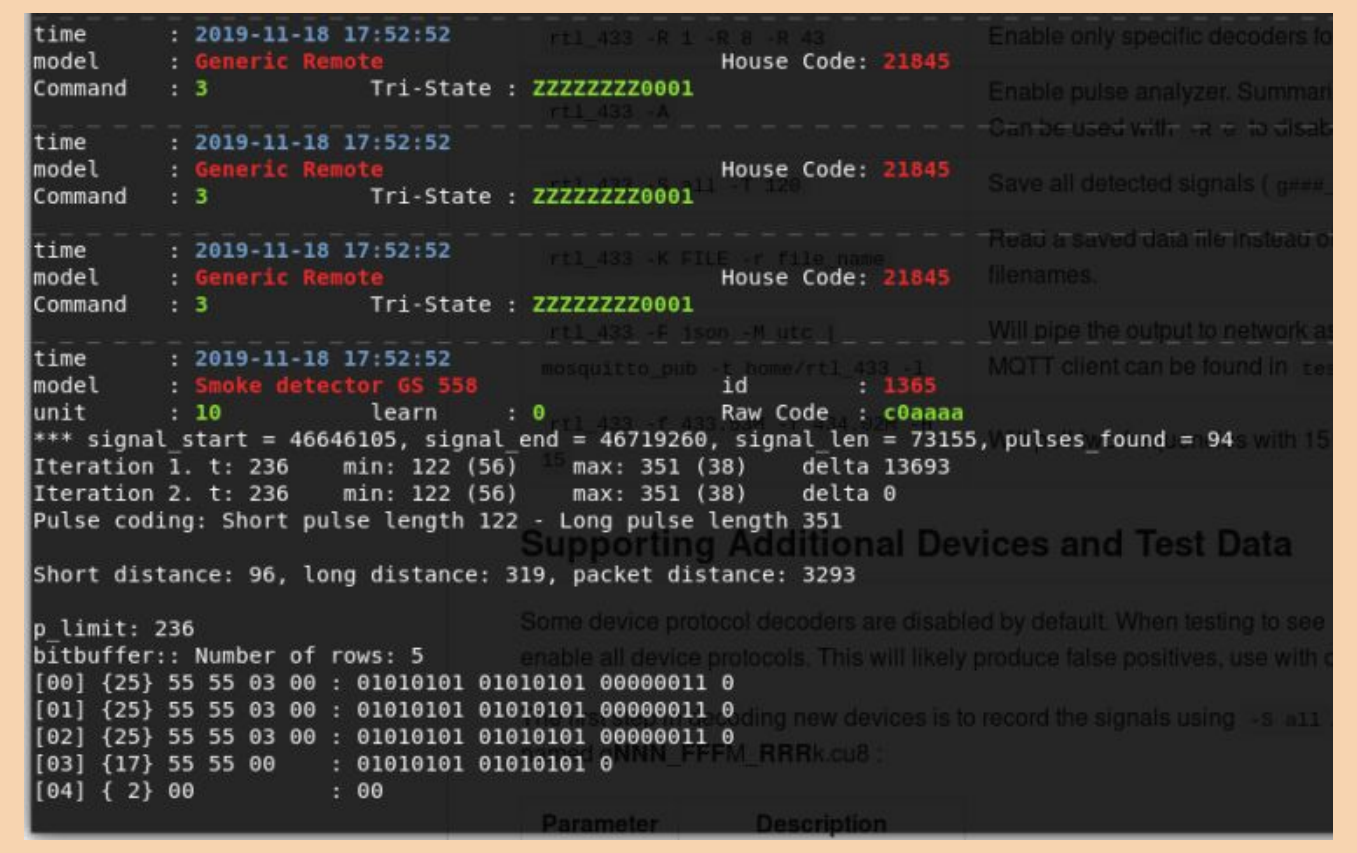
Results



Live RF Signal from Adafruit Key Fob seen from SDRSharp

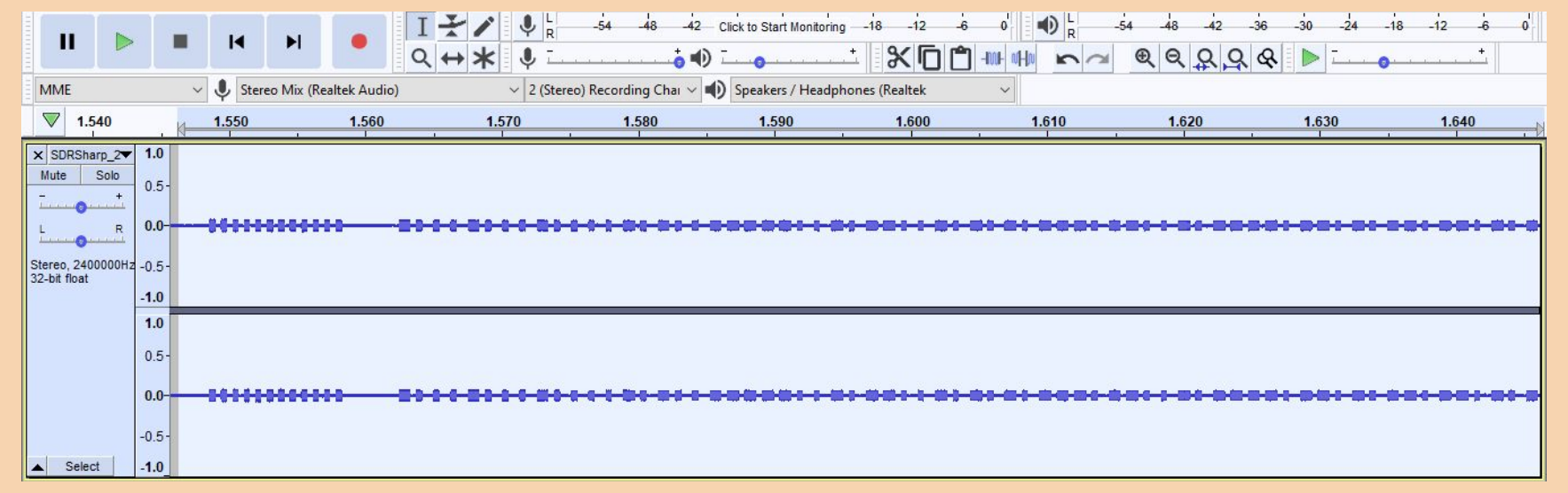


Data from 315 Mhz generic key fob in Audacity

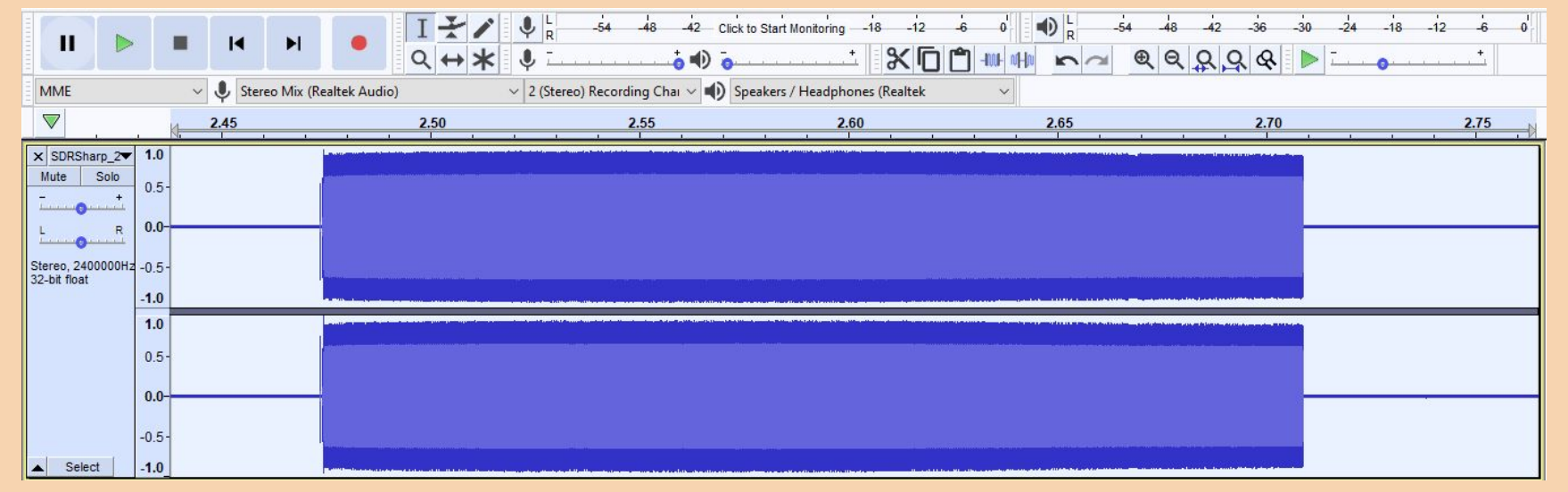


```
time      : 2019-11-18 17:52:52
model     : Generic Remote                      House Code: 21845
Command   : 3            Tri-State : ZZZZZZZZ0001
time      : 2019-11-18 17:52:52
model     : Generic Remote                      House Code: 21845
Command   : 3            Tri-State : ZZZZZZZZ0001
time      : 2019-11-18 17:52:52
model     : Generic Remote                      House Code: 21845
Command   : 3            Tri-State : ZZZZZZZZ0001
time      : 2019-11-18 17:52:52
model     : Smoke detector GS 558               id         : 1365
unit      : 10           learn     : 0          Raw Code   : c0aaaa
*** signal_start = 46646105, signal_end = 46719260, signal_len = 73155, pulses_found = 94
Iteration 1. t: 236    min: 122 (56)    max: 351 (38)    delta 13693
Iteration 2. t: 236    min: 122 (56)    max: 351 (38)    delta 0
Pulse coding: Short pulse length 122 - Long pulse length 351

Short distance: 96, long distance: 319, packet distance: 3293

p_limit: 236
bitbuffer:: Number of rows: 5
[00] {25} 55 55 03 00 : 01010101 01010101 00000011 0
[01] {25} 55 55 03 00 : 01010101 01010101 00000011 0
[02] {25} 55 55 03 00 : 01010101 01010101 00000011 0
[03] {17} 55 55 00    : 01010101 01010101 0
[04] { 2} 00          : 00
```

RTL_433 library used within Linux terminal to decode signals



Data from 315 Mhz car fob in Audacity



Data from 433 Mhz car fob in Audacity

References

1. Cartwright, Anthony. "TUTORIAL: How to Set up Wireless RF (433Mhz) Transmitter Receiver Module - Arduino Quick Simple." *YouTube*, YouTube, 30 June 2010, https://www.youtube.com/watch?v=KA_YE7AvFn0&t=831s.
2. "How to Copy a 433MHz Signal with an Arduino Board." YouTube, YouTube, 4 Mar. 2017, https://www.youtube.com/watch?v=LbCDpbWrdlQ.
3. Pendergast, Robert L., et al. "RF 433MHz Transmitter/Receiver Module With Arduino." *Random Nerd Tutorials*, 2 Apr. 2019, https://randomnerdtutorials.com/rf-433mhz-transmitter-receiver-module-with-arduino/.
4. Instructables. "Hack Remote RF Security Locks With Arduino." *Instructables*, Instructables, 22 Sept. 2017, https://www.instructables.com/id/Hack-Remote-RF-Security-Locks-With-Arduino-and-Usb/.

Acknowledgements

We would like to extend a huge thank you to Sparkdev and UPE for giving us the opportunity and resources to accomplish our project.