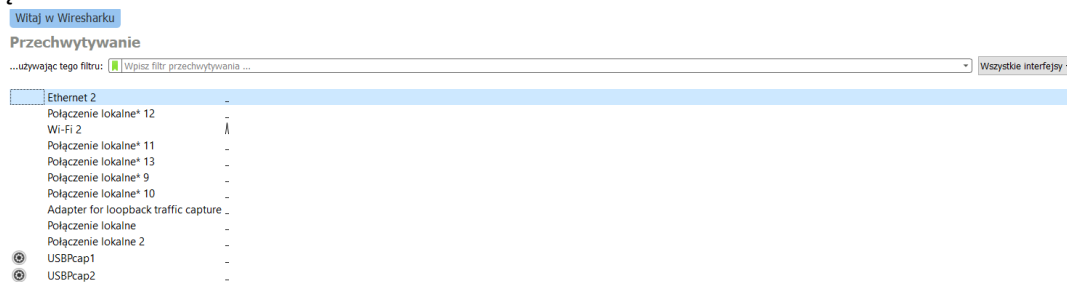


## Instrukcja:

### 1. Instalacja wtyczki USBPcap

- Wchodzimy na stronę <https://desowin.org/usbpcap/> i pobieramy instalator **USBPcapSetup-1.5.4.0.exe**.
- Następnie instalujemy z zachowaniem domyślnych ustawień i resetujemy komputer.
- Po resecie odnajdujemy folder z rozszerzeniami Wireshark-a:  
uruchamiamy Wireshark->Pomoc->O programie Wireshark->Foldery->odnajdujemy Global Extcap path i klikamy dwukrotnie, otworzy się folder
- Z folderu gdzie zainstalowaliśmy uprzednio wtyczkę USBPcap (u mnie C:\Program Files\USBPcap) kopiujemy plik USBPcapCMD.exe i wklejamy do folderu Global Extcap path.

Teraz po ponownym uruchomieniu Wiresharka powinniśmy zobaczyć możliwość sniffowania urządzeń USB:



### 2. Sniffowanie pakietów USB

- Klikamy USBPcap1 i obserwujemy czy po poruszaniu myszką uzyskujemy pakiety, jeśli nie uruchamiamy USBPcap2.
- Jako filtr wklejamy:

((usb.transfer\_type == 0x01) && (frame.len == 33)) && !(usb.capdata == 00:00:00:00:00:00:00:00)

Powinniśmy otrzymać podobny efekt:

((usb.transfer_type == 0x01) && (frame.len == 33)) && !(usb.capdata == 00:00:00:00:00:00:00:00)						
No.	Time	Source	Destination	Protocol	Length	Info
2545	23.457103	1.1.1	host	USB	33	URB_INTERRUPT in
2547	23.465102	1.1.1	host	USB	33	URB_INTERRUPT in
2549	23.473121	1.1.1	host	USB	33	URB_INTERRUPT in
2551	23.481083	1.1.1	host	USB	33	URB_INTERRUPT in
2553	23.489101	1.1.1	host	USB	33	URB_INTERRUPT in
2555	23.497118	1.1.1	host	USB	33	URB_INTERRUPT in
2557	23.505117	1.1.1	host	USB	33	URB_INTERRUPT in
2559	23.513118	1.1.1	host	USB	33	URB_INTERRUPT in
2561	23.521092	1.1.1	host	USB	33	URB_INTERRUPT in
2563	23.529107	1.1.1	host	USB	33	URB_INTERRUPT in
2565	23.537108	1.1.1	host	USB	33	URB_INTERRUPT in
2567	23.553127	1.1.1	host	USB	33	URB_INTERRUPT in

> Frame 7: 33 bytes on wire (264 bits), 33 bytes captured (264 bits) on interface wireshark\_extcap1076, id 0

> USB URB

HID Data: 0100ff0f0000

- Teraz klikamy prawym przyciskiem myszy na HID Data lub Leftover Capture Data i wybieramy Utwórz kolumnę z pola.

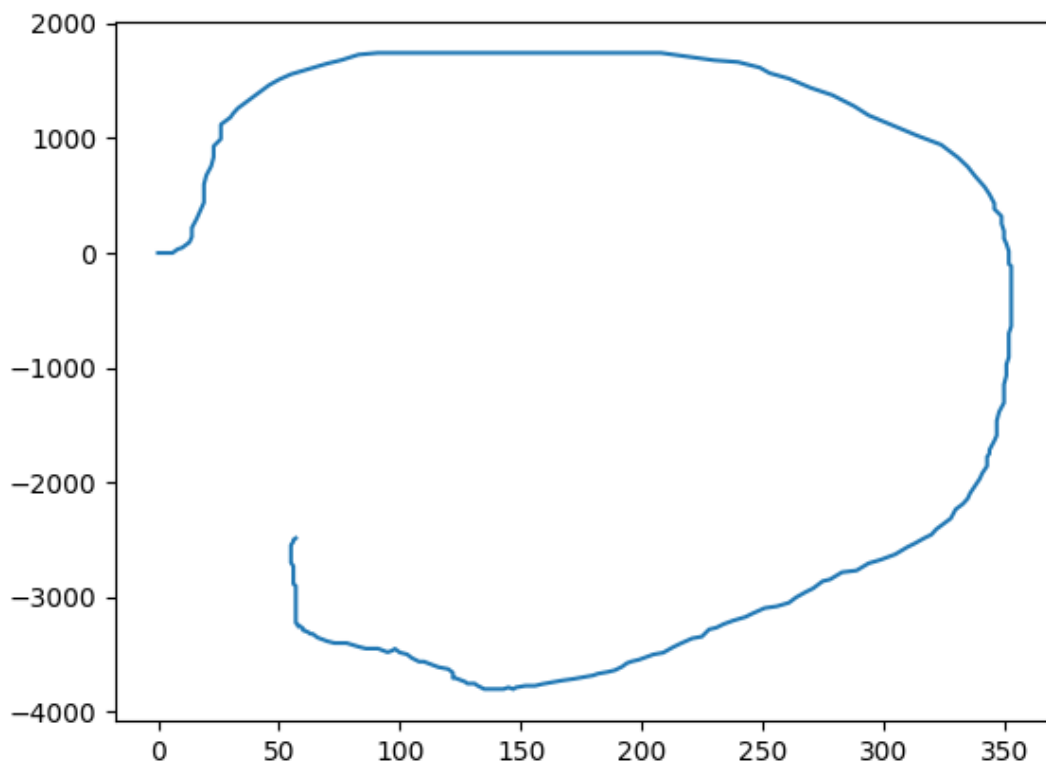
(((usb.transfer_type == 0x01) && (frame.len == 33)) && !(usb.capdata == 00:00:00:00:00:00:00))							
No.	Time	Source	Destination	Protocol	Length	HID Data	Info
13867	185.785186	1.1.1	host	USB	33	010003c00000	URB_INTERRUPT in
13869	185.793179	1.1.1	host	USB	33	010001f00000	URB_INTERRUPT in
13871	185.801179	1.1.1	host	USB	33	010002b00000	URB_INTERRUPT in
13873	185.809199	1.1.1	host	USB	33	010003c00000	URB_INTERRUPT in
13875	185.817305	1.1.1	host	USB	33	010001700000	URB_INTERRUPT in
13877	185.825195	1.1.1	host	USB	33	010000600000	URB_INTERRUPT in
13879	185.833203	1.1.1	host	USB	33	010002700000	URB_INTERRUPT in
13881	185.841167	1.1.1	host	USB	33	010000100000	URB_INTERRUPT in
13883	185.849199	1.1.1	host	USB	33	010000100000	URB_INTERRUPT in
13885	185.857238	1.1.1	host	USB	33	010000200000	URB_INTERRUPT in
13887	185.889195	1.1.1	host	USB	33	010001100000	URB_INTERRUPT in

> Frame 7: 33 bytes on wire (264 bits), 33 bytes captured (264 bits) on interface wireshark\_extcap1076, id 0  
 > USB URB  
 HID Data: 0100ff0f0000

Po uzyskaniu tego kroku możemy już zdekodować otrzymywane dane.

### 3. Dekodowanie ruchów myszki

- Z repozytorium: kopiujemy kod do VS Code i zapisujemy, najlepiej w nowym folderze na pulpicie.
- Wracamy do Wiresharka, uruchamiamy przechwytywanie pakietów i wykonujemy ruch myszką, na przykład poruszamy nią „rysując” koło.
- Stopujemy przechwytywanie pakietów i wybieramy Plik -> Eksportuj prezentację pakietów -> jako CSV.
- Zapisujemy plik csv pod nazwą sisk(można użyć oczywiście innej, wiąże się to tylko ze zmianą nazwy w kodzie) w folderze, gdzie wcześniej zapisaliśmy kod pythona.
- Uruchamiamy kod w VS Code, powinniśmy otrzymać zdekodowane ruchy myszki:



Dodatkowo jeśli w trakcie rysowania trzymaliśmy jakiś przycisk myszy, to pojawi się to w logach.