**[컴퓨터 네트워크]**

**CRC 체크 프로그램 개발**

* 20173158 조영재 –

**개발 언어 : Python3.10**

**개발 환경 : Mac**

**편집 도구 : Visual Studio Code**

**GitHub URL: https://github.com/co2plant/CRC\_Checking**

텍스트이(가) 표시된 사진

자동 생성된 설명

[ 메인이 되는 소스코드(main\_String.py) ]

1. 프로그램 시작 시 제수, 페이로드 패킷(데이터), 이니셜 코드를 문자열로 입력 받습니다.

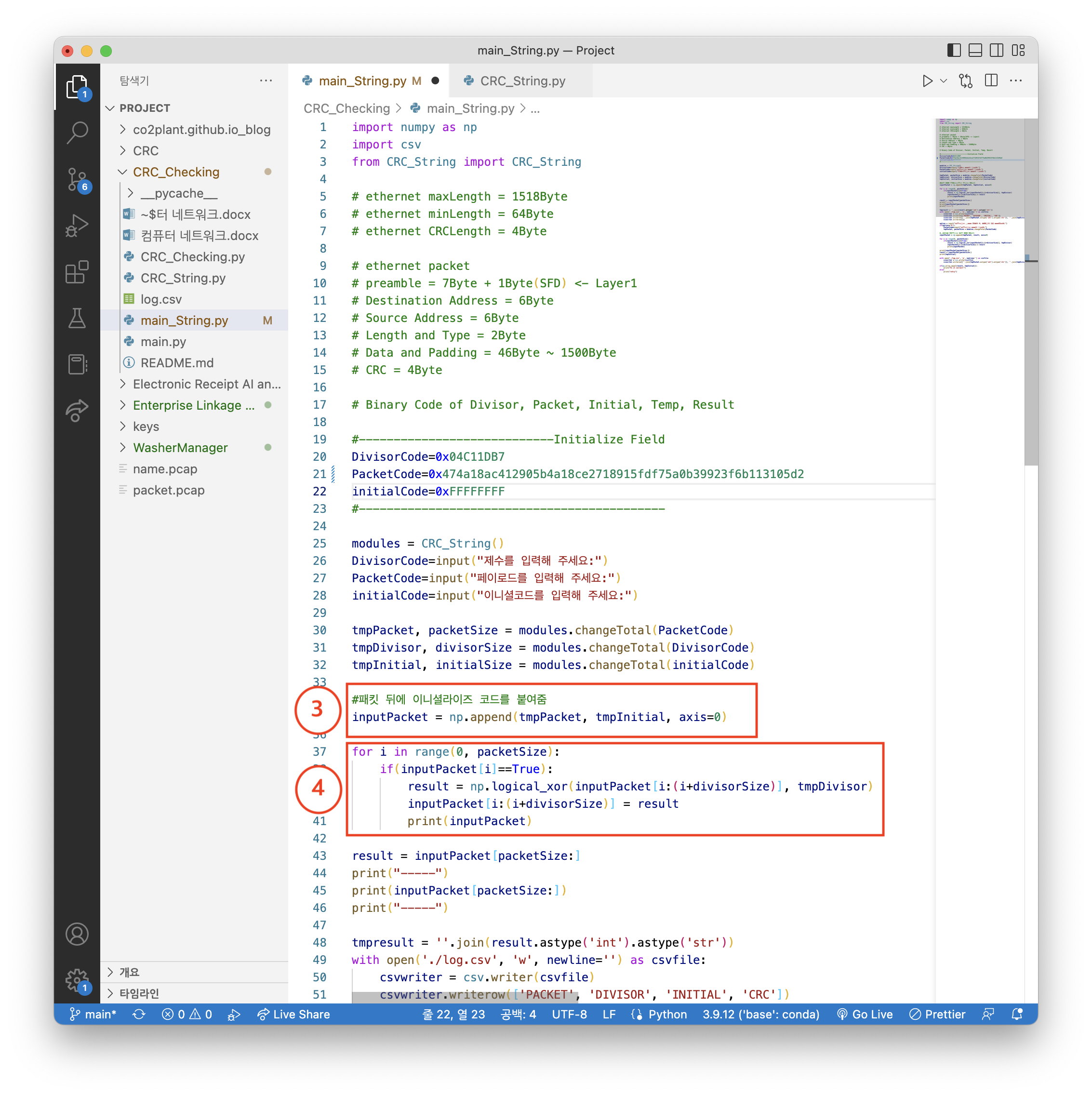
이때, 이니셜 코드는 CRC마다 달라서 임의대로 넣을 수 있도록 했습니다.

예를 들어 CRC-32를 사용하는Ethernet에서 이니셜 코드는 0xFFFFFFFF가 됩니다.

2. 입력 받은 3개의 문자열들을 계산을 위해 Boolean형의 numpy배열로 바꿔줘야 합니다. 미리 정의한 CRC\_String.py의 changeTotal()을 이용해 바꿔줍니다.

텍스트이(가) 표시된 사진

자동 생성된 설명



3. numpy배열로 변환된 packetCode와 InitialCode를 append()함수로 붙여 계산할 때 사용할 inputPacket을 만들어 줍니다.

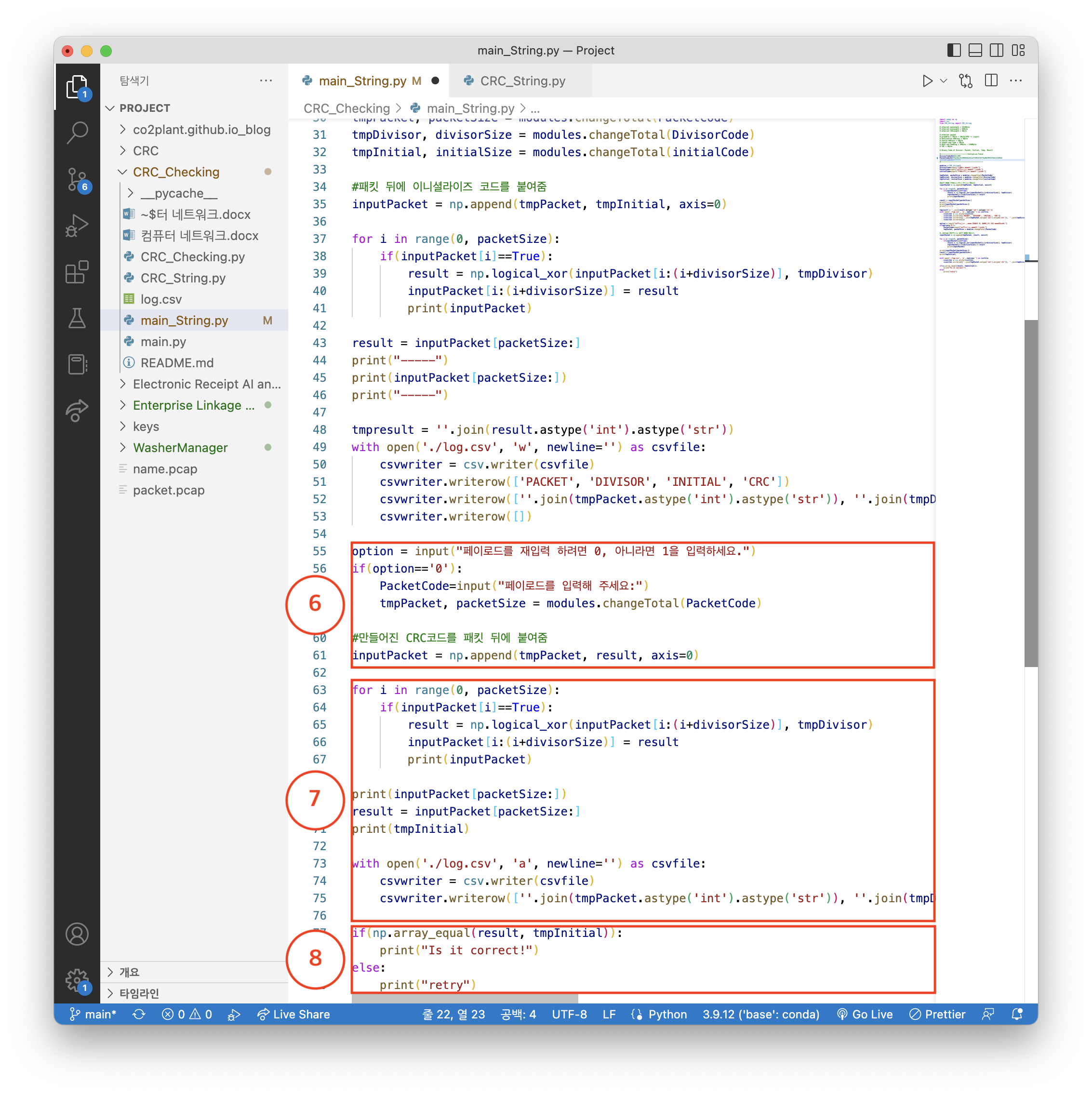
4. 반복문은 0~packetSize-1만큼 돌아가며, 이때 뒤에 initialCode길이만큼 CRC 코드를 남기기 위해 접합한 inputPacket길이가 아닌 packetCode의 길이까지만 나눠줍니다.

inputPacket[i]가 1이면 inputPacket[i:i+divisorSize]와 divisorCode를 논리합 연산을 하면서 CRC코드를 만들게 됩니다.

텍스트, 모니터, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

5. 원래는 콘솔로만 값을 보여줬으나 보기 힘든 관계로 CSV파일을 이용해서 결과를 보여주도록 변경해봤습니다.



6. 이후 페이로드(데이터)가 깨지는 상황을 연출하기 위해서 0을 입력하면 재입력을 통해 깨진 데이터를 넣어볼 수 있게 했습니다.

7. 페이로드(데이터)만 변화를 준 채로 4번해서 했던 연산을 똑같이 진행하고, 5번에서 저장했던 것처럼 CSV파일에 결과를 저장해 줍니다.

8. 7번의 연산 결과인 result와 알고리즘 내에서 사용한 initialCode가 같은 지 비교하고 같다면 깨짐이 없는 것이고, 다르다면 데이터가 깨졌다는 것을 알 수 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[ 실제로 캡쳐된 UDP packet ]

실제로 캡쳐된 UDP packet, 26Byte길이의 Data를 이용해 CRC check를 해봤습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[ CSV에 저장된 결과 ]

사용한 Data는 0x474a18ac412905b4a18ce2718915fdf75a0b39923f6b113105d2

제수는 0x04C11DB7

이니셜 코드는 0xFFFFFFFF를 사용했으며 중간에 데이터가 깨진 것을 연출하기 위해 Data를 0x474a18ac412905b4a18ce2718915fdf75a0b39923f6b113105d6

으로 바꿨습니다.

10번째 줄의 CRC를 보면 이니셜 코드인 0xFFFFFFFF와 다르게 중간에 0이 다수 껴있는걸 볼 수 있는데 이것을 통해 데이터가 깨졌다는 것을 알 수 있습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

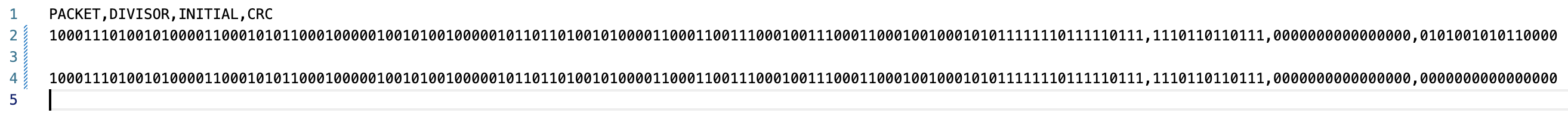
[ 콘솔에 띄워진 결과 ]

콘솔에도 데이터가 깨졌다는 것을 알려주는 ‘retry’를 출력하며 프로그램은 종료됩니다.



[ 16Byte의 데이터 코드와 임의의 CRC-16 ]

다른 결과를 보기 위해서 initialCode도 0x0000으로 만들고 나머지는 길이를 잘라보았습니다.

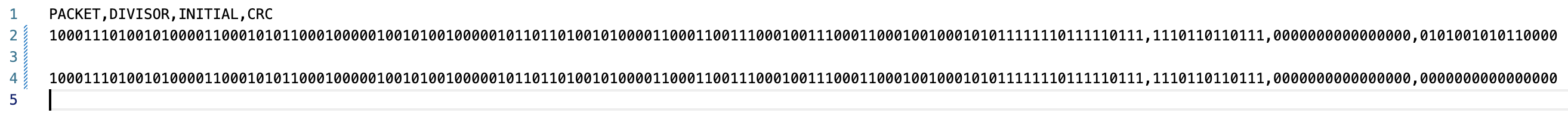


[ CSV에 저장된 결과 ]

이번에는 올바르게 전달된 상황을 가정해서 packet코드를 건들지 않았습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

[ 결과 화면 ] 

[ 좌측부터 제수, 이니셜코드, CRC코드 ]

결과 화면에도 문제 없다고 출력됐으며 CSV에 CRC코드도 이니셜 코드와 똑같이 0000000000000000으로 통일된 것을 확인 할 수 있습니다.

**참조**

CRC-32의 initial Code와 divisorCode

[ https://crccalc.com/ ]

* **감사합니다 -**