TCP_middle.py

```
# TCP 소켓 생성
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
    try:
        # 서버와 연결
        sock.connect(server_address)

        # 메시지 전송을 30분 동안 계속 진행
        while time.time() < end_time:
            for hl7_message in hl7_messages:
                # 메시지 전송
                sock.sendall(hl7_message.encode('utf-8'))
                print(f"Sent: {hl7_message}")  # 전송한 메시지 출력

                time.sleep(60) #<-- 파라미터 설정

            # 메시지 전송이 끝나면 종료 조건을 체크
            if time.time() >= end_time:
                break  # 30분이 지났으면 종료

    except Exception as e:
        print(f"Error: {e}")

  return go(f, seed, [])
}
```

시간(s) 설정

test_middle.py

```
if field_description == "new_value":
                    x_value, y_value = field_str.split(',')

                    if x_value.isdigit():  # 숫자인지 확인
                        x_value = int(x_value)

                        if message_counter < one_round:  # 첫 번째 주기
                            save_X_value = x_value  # x 값 그대로 사용

                        else:  # 두 번째 이후부터는 1초마다 10씩 증가
                            save_X_value += time_second  # 1초마다 10씩 증가
                            x_value = save_X_value
                if field_description == "new_value":
                    segment_dict["new_value"] = f"{x_value},{y_value}"

                if field_description == "elapsedTime":
                    segment_dict["elapsedTime"] = x_value


                if field_description == "metadata":
                    metadata = {}  # 새로운 setting 초기화
                    metadata['x'] = x_value
                    metadata['y'] = y_value

                    if metadata:
                        segment_dict["metadata"] = metadata

            if segment_dict:  # 빈 세그먼트가 아니면
                analyzed_fields.append({segment_code: segment_dict})
```

1 주기를 돌면 x값이 변한다

X값 변화

값 저장

test_middle.py

```
save_X_value = True
time_second = 10 * 60   # 시간 설정
message_counter = 0
one_round = 5    # 데이터 개수
```

시간 설정

데이터 개수 설정

test_middle.py

```python
def start_server():
    server_address = ('localhost', 12345)  # 서버의 주소와 포트
    # TCP 소켓 생성
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
        try:
            # 서버와 연결
            sock.bind(server_address)
            sock.listen(1)  # 클라이언트의 연결을 기다림
            print(f"Server is waiting for connections on {server_address}...")

            # 연결 대기
            connection, client_address = sock.accept()
            with connection:
                print(f"Connection established with {client_address}")

                # 클라이언트로부터 메시지를 받는 부분
                while True:
                    data = connection.recv(2048)  # 2048바이트씩 수신
                    if not data:
                        break  # 더 이상 데이터가 없으면 종료
                    hl7_message = data.decode('utf-8')  # 수신된 메시지 디코
                    #print(f"Received: {hl7_message}")  # 수신된 메시지 출력

                    analyze_segment(hl7_message)

                    time.sleep(5)  # 10초 대기
        except Exception as e:
            print(f"Error: {e}")
```

시간(s) 설정