

스크립트를 사용한 동적 프로토콜 통신 개발에 관한 연구

김한승
(주)효성 중공업 연구소

The Study on Dynamic Protocol Communication Development Which Uses a Script

Han-Sung Kim
Hyosung Corporation Power & Industrial System R&D Center

Abstract - 수 처리 분야의 통합시스템을 구현함에 있어 이기종 장비 간의 데이터 연계는 프로그래머의 통신 프로그래밍 작업으로 매번 개발을 하여하는 문제점을 가지고 있다. 이로 인해 시스템의 안정화에 많은 시간 및 인력이 투입되고 있다. 본 연구는 인터프리터 엔진을 활용하여 엔지니어가 엔지니어링 한 데이터(통신 스크립트)를 번역, 해석, 자동동작 시켜 별도의 프로그래밍 작업(개발)없이 엔지니어링으로만 통합 시스템의 감시 및 제어를 구축 가능케 프로토콜 편집기를 개발 하여 신뢰성 높은 시스템을 구축키 위한 내용이다.

1. 서 론

최근 10년간 IT기술의 눈부신 발전은 우리 사회를 크게 바꿔 놓았다. 이는 산업 분야에도 예외는 아니어서 산업현장의 곳곳이 IT 기술과 접목되고 있다. 프로세스 제어분야에서도 많은 변화가 있었는데, PC 기반의 HMI 구현기술, 웹을 통한 감시제어 기술, 통합 시스템 기술 등이 새로 개발되거나 제안되었다.

본 논문은 프로세스 제어분야에서는 잘 다루어지지 않은 내용을 소개한다. 소켓통신 함수를 지원하는 스크립트 언어를 사용하여 통신 프로토콜을 구성하는 내용이다. 또한 이러한 활동으로 인해 얻을 수 있는 이익과 개발의 의의에 대해 알려려고 한다.

2. 본 론

2.1 개요

본 논문은 스크립트 엔진(인터프리터)과 그 엔진을 적용한 간단한 프로그램의 개발과 활용에 대해서 설명을 한다. 그리고 이것들을 이용하여 통신 스크립트 샘플 코드를 작성하고 시뮬레이션 해본다.

먼저 개발한 스크립트 엔진이 지원하는 기능에 대해서 설명하고, 본사에서 개발할 프로그램의 프로토타입인 스크립트 편집기 프로그램의 기능과 사용법을 소개한다. 마지막으로 실제로 소켓통신을 하는 스크립트 코드를 작성하고 시뮬레이션 해 볼 것이다.

2.2 스크립트 엔진

2.2.1 기능

본사에서 개발한 스크립트 엔진(인터프리터)이 현재 지원 가능한 통신관련 함수는 (표1)과 같다.

함수	항목	Prototype	Argument	Return	Description
Socket Init		SocketInit() ()	-	성공:0 실패:1	socket 초기화
Socket Open		SocketOpen() ()	-	성공:소켓디스크립터 실패:-1	socket 생성
Socket Bind		SocketBind(int hSocket, int port) (int hSocket, int port)	hSocket:소켓디스크립터 port:연결할 port번호	성공:-1외의 수 실패:-1	TOP 연결에 사용되는 LocalPort 및 LocalIP를 지정
Socket Listen		SocketListen(int hSocket, int maxCIntSocket) (int hSocket, int maxCIntSocket)	hSocket:소켓디스크립터 maxCIntSocket:최대 클라이언트의 개수	성공:-1외의 수 실패:-1	클라이언트의 접속을 기다림

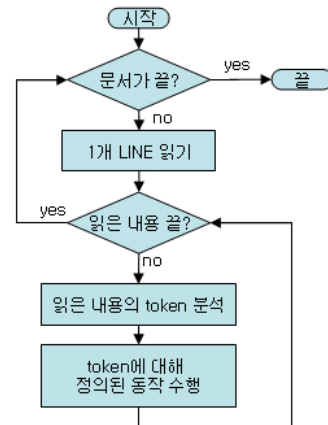
SocketAccept	SocketAccept(int hSocket) (int hSocket)	hSocket:소켓디스크립터	성공:소켓디스크립터 실패:-1	클라이언트의 Connect 요청을 받으면 이를 받아들여 클라이언트 socket생성
SocketConnect	SocketConnect(int hSocket, char *ip, int port) (int hSocket, char *ip, int port)	hSocket:소켓디스크립터 ip:서버 ip port:서버 port	성공:-1외의 수 실패:-1	서버측에 연결을 요청
SocketSend	SocketSend(int hSocket, void *sendData, int sendSize) (int hSocket, void *sendData, int sendSize)	hSocket:소켓디스크립터 sendData:전송할 데이터 sendSize:데이터의 크기	성공:0 실패:-1	데이터 송신
SocketRecv	SocketRecv(int hSocket, void *recvData, int recvSize) (int hSocket, void *recvData, int recvSize)	hSocket:소켓디스크립터 recvData:전송받은 데이터 recvSize:데이터의 크기	성공:전달받은 값 실패: "SocketRecv Error."	데이터 수신
SocketClose	SocketClose(hSocket) (hSocket)	hSocket:소켓디스크립터	0	socket 종료

〈표1, 소켓통신 관련함수 목록〉

(표1)에 설명된 함수 이외에도 사칙연산, 조건문, 메모리 관리 등의 기본적인 기능들이 포함되어 있다. 또한 기능 확장이 필요할 경우에는 함수의 추가가 용이하게 설계되어 있다.

2.2.1 동작

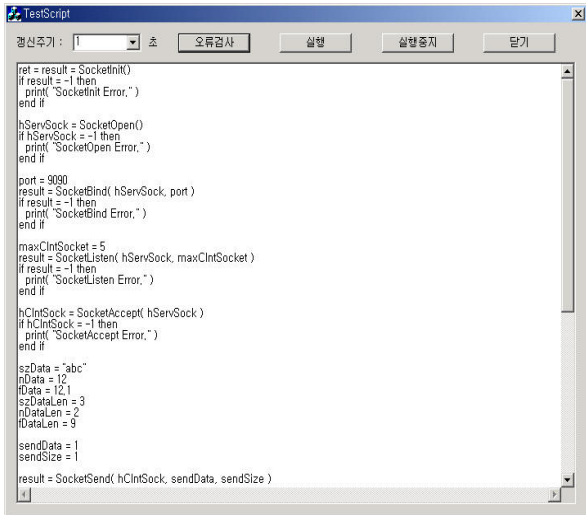
아래 (그림1)은 본 논문에서 다루는 스크립트 엔진의 해석과 실행에 대한 간단한 순서도이다.



〈그림 1〉 스크립트 엔진 실행 순서도

2.3 스크립트 편집기

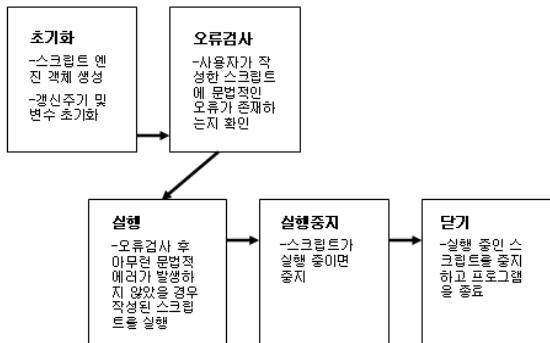
2.3.1 기능 및 동작



〈그림 2〉 스크립트 편집기 화면

(그림2)는 스크립트 엔진을 사용한 스크립트 편집기 화면이다. 텍스트 편집기를 사용해 스크립트를 작성 할 수 있고, 스크립트 실행과 오류검사를 지원한다. 또한 실행 주기를 지정할 수 있다. 아래는 각 버튼 기능에 대한 설명이다.

- * 갱신주기
 - 스크립트가 실행되는 주기를 설정
 - 초단위로 설정 가능
- * 오류검사
 - 사용자가 작성한 스크립트에 문법적인 오류가 존재하는지 확인
 - 오류가 존재하는 줄(Line)을 표시
- * 실행
 - 오류검사를 수행하고 아무런 문법적 오류가 탐지되지 않았을 경우 작성된 스크립트를 실행
- * 실행 중지
 - 스크립트가 실행 중이면 중지
- * 닫기
 - 실행 중인 스크립트를 중지하고 프로그램을 종료



〈그림 3〉 스크립트 편집기 동작 순서도

스크립트 편집기의 동작은 (그림3)과 같은 단계로 수행된다. 먼저 '실행' 버튼을 누르면 스크립트를 해석하고 동작하는 엔진 객체와 갱신주기 및 관련 설정들을 초기화 한다. 그리고 작성된 스크립트에 대해 오류검사를 수행하고 오류가 있으면 경고를 한 후 동작을 멈춘다. 오류가 없을 경우 초기화시 설정된 갱신주기대로 반복동작 한다. '실행 중지' 버튼을 누르면 스크립트 동작이 멈추게 된다.

2.3.2 통신 스크립트 활용 예

다음은 스크립트를 사용한 소켓통신의 예이다. 두 대의 PC에서 각각 스크립트 편집기를 설치하고 아래와 같은 두 가지 스크립트를 각각 작성하고 실행했다.

- * 테스트용 스크립트 소스 (서버)
 - 클라이언트가 서버에 접속하면 '1'이라는 데이터를 클라이언트 측에 전송하고 접속을 끊는다.

```

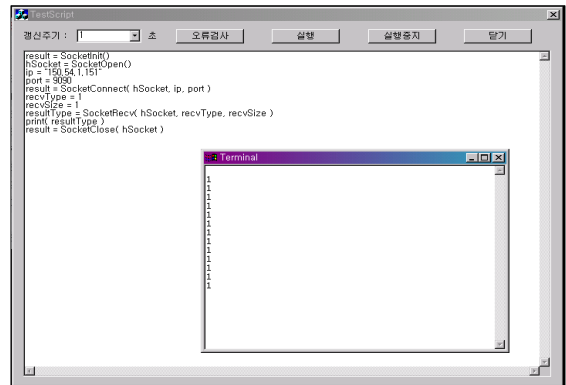
result1 = SocketInit()
hServSock = SocketOpen()
port = 9090
result2 = SocketBind( hServSock, port )
maxCntSocket = 5
result3 = SocketListen( hServSock, maxCntSocket )
hClnSocket = SocketAccept( hServSock )
sendData = 1
sendSize = 1
result4 = SocketSend( hClnSocket, sendData, sendSize )
result = SocketClose( hClnSocket )
  
```

- * 테스트용 스크립트 소스 (클라이언트)
 - 서버 측에 접속하고 받은 데이터를 출력한 후 접속을 끊는다

```

result = SocketInit()
hSocket = SocketOpen()
ip = "150.54.1.155"
port = 9090
result = SocketConnect( hSocket, ip, port )
recvType = 1
recvSize = 1
resultType = SocketRecv( hSocket, recvType, recvSize )
print( resultType )
result = SocketClose( hSocket )
  
```

- * 실행 결과
 - (그림4)에서 보는 것과 같이 클라이언트에서 '1'이라는 메시지가 반복 출력되었다.



〈그림 4〉 통신 스크립트샘플 수행 결과 화면

2.3.3 개발의 의의

스크립트를 사용한 동적 프로토콜 구현 기술은 수 처리 분야의 통합 시스템을 구현함에 있어서 통신을 통한 이기종간의 데이터 연계를 용이하게 할 목적을 가지고 있다. 또한 기존에 존재하는 H/W제품을 대체하려는 목적도 가지고 있다. 본 논문에서 다른 프로그램의 제품화가 이루어지게 되면 PLC같은 형태의 장비에 대해서 데이터 취득이 용이하게 된다. 각 메이커의 전용 장비 없이 데이터를 취득할 수가 있는 것이 큰 장점이다. 또한 PLC에 대한 데이터 취득뿐만 아니라 2.3.2절의 예에서와 같이 자체적으로 서버-클라이언트로 동작할 수도 있다.

3. 결 론

본 논문은 스크립트 언어를 사용해서 통신 프로토콜을 구현하고 인터프리터 엔진으로 구동하는 개념의 연구에 대한 소개하고 있다. 기존의 서버-클라이언트 시스템에서 좀 더 발전된 개념으로서 통신서버 프로그램의 범용성과 재사용성을 추구한다는 점에서 의의를 가진다. 본 연구에서는 프로그래밍 언어의 한 종류인 인터프리터 언어를 통해 통신프로토콜을 구현하는 시도를 해 보았다. 결과는 희망적이었으며 앞으로 제품화 할 예정이다 있다.

최근 계장 시스템 분야와 IT기술의 접목이 급격하게 진행되고 있다. 당사는 현재 이러한 요구에 대해 빠르게 적응하고 있으며 앞으로 환경 변화에 적극적으로 대응하기 위해 노력하고 있다.

[참 고 문 헌]

[1] (주)효성, "통합용 게이트웨이 서버 기술매뉴얼", 2007년
 [2] (주)효성, "비 웹 통합 HMI 제품설명서", 2006년