SENA
TECHNOLOGIES

Starter's Kit for HelloDevice 1100

사용자 매뉴얼

Version 1.1

목 차

1. 시작하기 전에

▶ HelloDevice 1100 Starter's Kit 공급 품목

품 목		개수
하드웨어	HelloDevice 1100 메인 보드	1
	5V 스위칭모드 파워서플라이 (SMPS) 또는 전원 어댑터	1
	입출력 제어용 데모보드	1
	입출력 제어용 케이블 (20 핀, 2.5mm 피치)	2
소프트웨어	HelloDevice 유틸리티 소프트웨어 및 데모용 예제 프로그램 (http://www.sena.com에서 최신 버전 다운로드)	1
사용자 설명서		1

▶ 사용 환경

- 공급 전압 = 5 V DC ±10%
- 공급 전류 = 200mA 이하
- 동작 습도 = 0 ~ 95%
- 동작 온도 = 섭씨 0 ~ 50 도

▶ 기술 지원

㈜세나테크놀로지
인터넷사업팀 기술지원담당
137-130 서울시 서초구 양재동 210
전화: (02) 573-7772
팩스: (02) 573-7710
Email : support@sena.com
http://www.sena.com

2. 개요

HelloDevice 는 원격으로 전자기기와 통신하여 그 기기를 제어하기 위한 솔루션으로, HelloDevice1x00 시리즈는 10 Base-T 이더넷 인터페이스를 통하여 네트워크에 연결되는 보드 형태의 제품군을 말한다. HelloDevice 1x00 시리즈는 하부 기기와 연결하는 방식에 따라,

① 디지털 입출력 방식 (HelloDevice 1100)
② 듀얼포트램 방식 (HelloDevice 1200),
③ 시리얼 포트 방식 (HelloDevice 1300)

으로 구분된다. HelloDevice 사용자는 마이크로소프트사의 익스플로러나 네트스케이프사의 네비게이터와 같은 표준 웹 브라우져를 통해 인터넷을 접속할 수 있는 어느 곳에서나 웹기반으로 기기를 감시 및 제어할 수가 있다.

HelloDevice 1x00 시리즈의 기본 사양은 아래의 표 2.1 과 같다.

모델 / 기본 사양	HelloDevice 1100	HelloDevice 1200	HelloDevice 1300
CPU	Scenix Sx52BD (8-Bit 마이크로프로세서, 50 MIPS)		
메모리	512 KB 플래쉬메모리 (사용자 웹 파일 저장)		
네트워크 연결	10 Base-T 이더넷 인터페이스 (IEEE802.3)		
하부 장치 연결	16 접점 디지털 입력 16 접점 디지털 출력	2 KB 듀얼포트램	1 채널 RS-232/485 최대 38400 bps
인터넷 프로토콜 지원	HTTP[1] / SMTP / BOOTP		
	TCP / UDP		
	IP / ICMP / ARP		
	이더넷 (IEEE802.3)		
유틸리티 소프트웨어	HelloDevice 유틸리티 소프트웨어 (윈도우즈 95/98/NT/2000 용) : IP 주소 설정, 웹 페이지 업로드		

표 2.1 HelloDevice 1x00 시리즈 기본 사양

[1] 현재는 HTTP 1.1 을 지원한다.

3. 메인보드 사양

HelloDevice 1100 의 하드웨어 구성과 시스템 블록도는 아래의 그림 3.1 과 그림 3.2 과 같다.

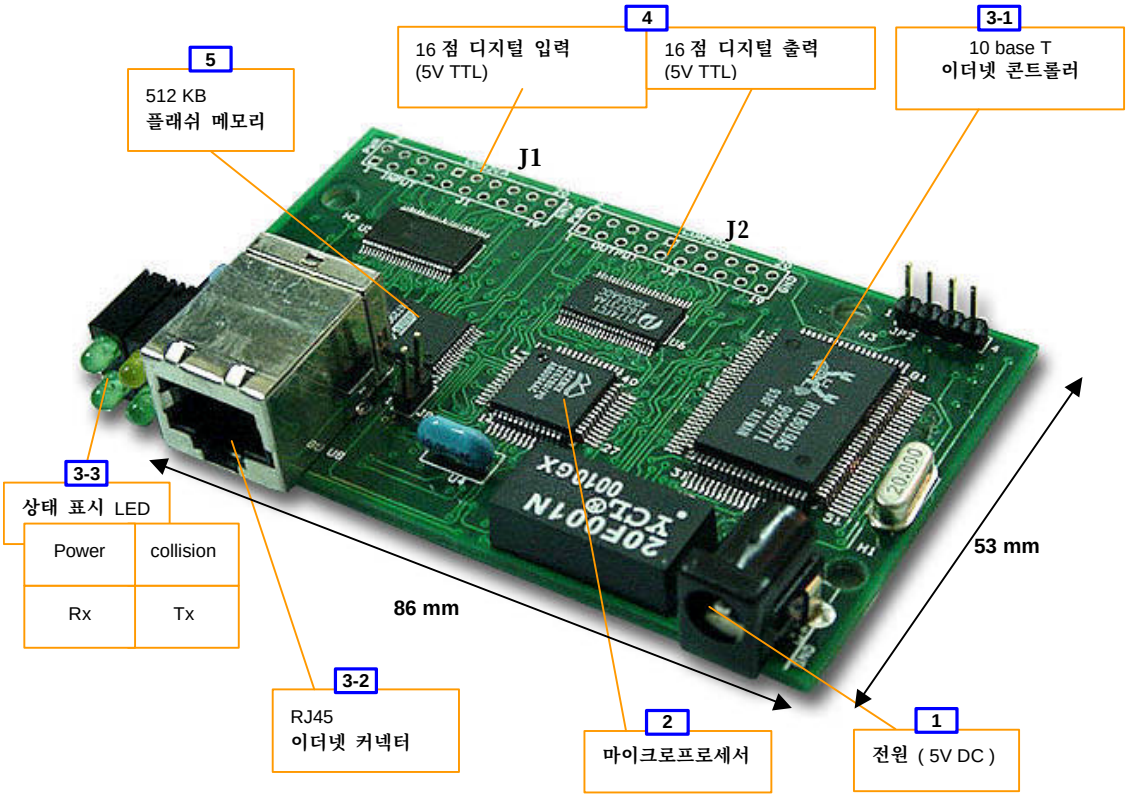


그림 3.1. HelloDevice 1100 구성 및 명칭

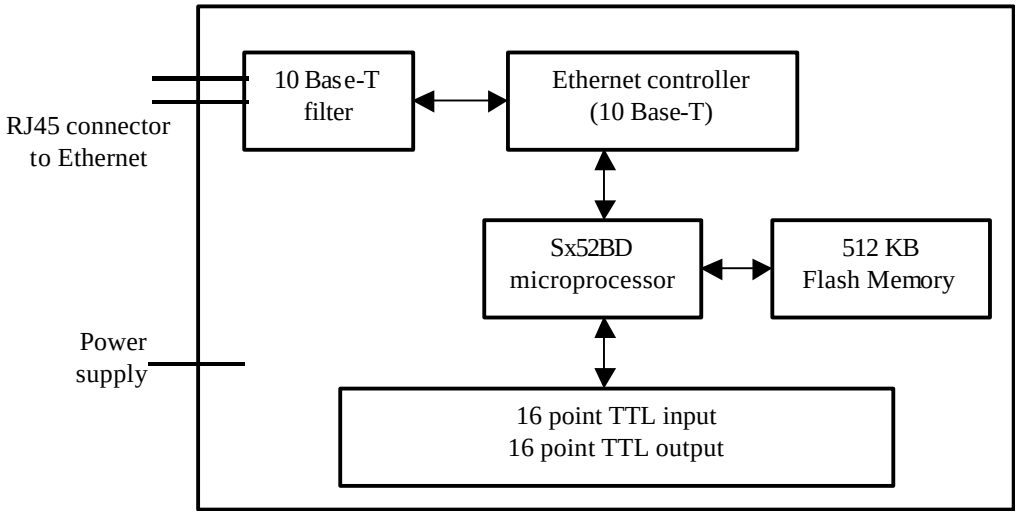


그림 3.2. HelloDevice 1100 시스템 블록도

3.1 전원

- 공급 전압 = 5 V DC ±10%
- 공급 전류 = 200mA 이하

3.2 마이크로프로세서 사양

- Scenix Sx52BD 8-bit 마이크로프로세서
- 4 KByte 프로그램 메모리
- 52 핀 PQFP (그림 3.1, 부품 [2] 참조)

3.3 이더넷 인터페이스

이더넷 케이블은 HelloDevice 의 RJ45 커넥터로부터 허브나 스위치에 연결하며, 최대 100m 까지 연결이 가능하다.

3.3.1 이더넷 콘트롤러

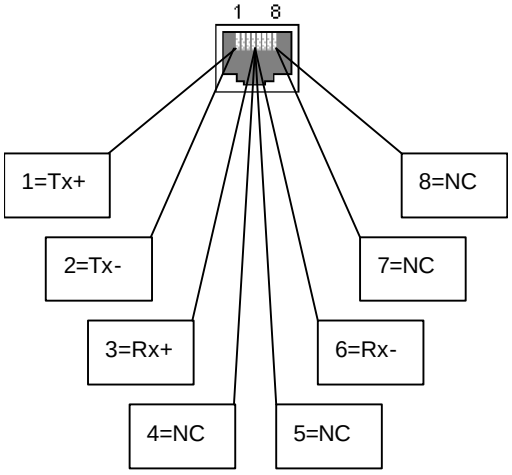
이더넷 콘트롤러에서는, 모든 데이터 프레이밍이나 어드레싱, 오류 및 충돌 감지, 회피 등의 기능을 수행한다 (그림 3.1, 부품 [3-1] 참조).

사 양:

- RealTek Full-duplex 이더넷 콘트롤러 : RTL8019AS
- IEEE802.3 10 base-5 , 10 base-2 , 10 base-T 지원
- 16 Kbyte SRAM
- NE2000 호환

3.3.2 RJ45 커넥터

- AT&T258 규격에 부합하는 Shield 처리 커넥터 (그림 3.1, 부품 [3-2] 참조)



Pin 번호	설명	전선색
1	Tx+	White with orange
2	Tx-	Orange
3	Rx+	White with green
4	Not used	Blue
5	Not used	White with blue
6	Rx-	Green
7	Not used	White with brown
8	Not used	Brown

그림 3.3 RJ45 커넥터

3.3.3 상태 표시 LED

상태 표시 LED 로는 Tx, Rx, Collision, Power LED 의 4 종류가 있으며(그림 3.1, 부품 [3-3] 참조), 그 용도는 다음과 같다.

- **Power LED**
  HelloDevice 전원 ON 상태를 나타냄.
- **Rx LED**
  이더넷 콘트롤러에서 이더넷 패킷을 네트워크로부터 수신하고 있음을 나타냄.
  정상적인 네트워크 연결 상태이면, 계속 점등/점멸을 반복하게 된다.
- **Tx LED**
  이더넷 콘트롤러에서 이더넷 패킷을 네트워크로 전송하고 있음을 나타냄.
  HelloDevice 네트워크 설정을 하기 전에는 1 초 주기로 계속 점등/점멸을 반복하게 된다.
  네트워크 설정 후, 정상적인 네트워크 연결 상태이면, ping, 서버 접속 등 사용자 PC 의 요구가 있을 경우에만 점등/점멸을 반복하게 된다.
- **Collision LED**
  이더넷 콘트롤러에서 네트워크로 전송한 이더넷 패킷이 다른 패킷과 충돌했음을 나타냄.
  이 경우, 이더넷 콘트롤러에서는 충돌 발생한 패킷을 자동으로 다시 전송하게 된다.

3.4 디지털 입출력 인터페이스

HelloDevice 1100 의 디지털 입출력 포트는, 각각 16 접점의 입출력 감지 및 제어 기능을 갖는다. 사용자는 스위치나 ON/OFF 센서 등을 입력 접점에 연결할 수 있으며, LED 나 ON/OFF 액튜에이터 등을 출력 접점에 연결할 수 있다. 입출력 제어를 위한 전압 사양은 다음과 같으며, 동작을 위한 전원은 HelloDevice 커넥터로부터 공급 받는다.

제어 신호	전압 레벨	최소 (V)	보통 (V)	최대 (V)
입력	입력 Hi 전압 (ViH)	2	5	
	입력 Low 전압 (ViL)		0	0.8
출력	출력 Hi 전압 (VoH)	2.5	5	
	출력 Low 전압 (VoL)			

표 3.1. HelloDevice 입출력 동작을 위한 전원 사양

사용자 장치와 HelloDevice 는 2.5mm 피치, 20 핀 커넥터 J1 과 J2 를 통하여 디지털 입출력이 이루어진다. 그림 3.4 에 각각의 입출력 커넥터에 따른 핀 할당을 나타내었다

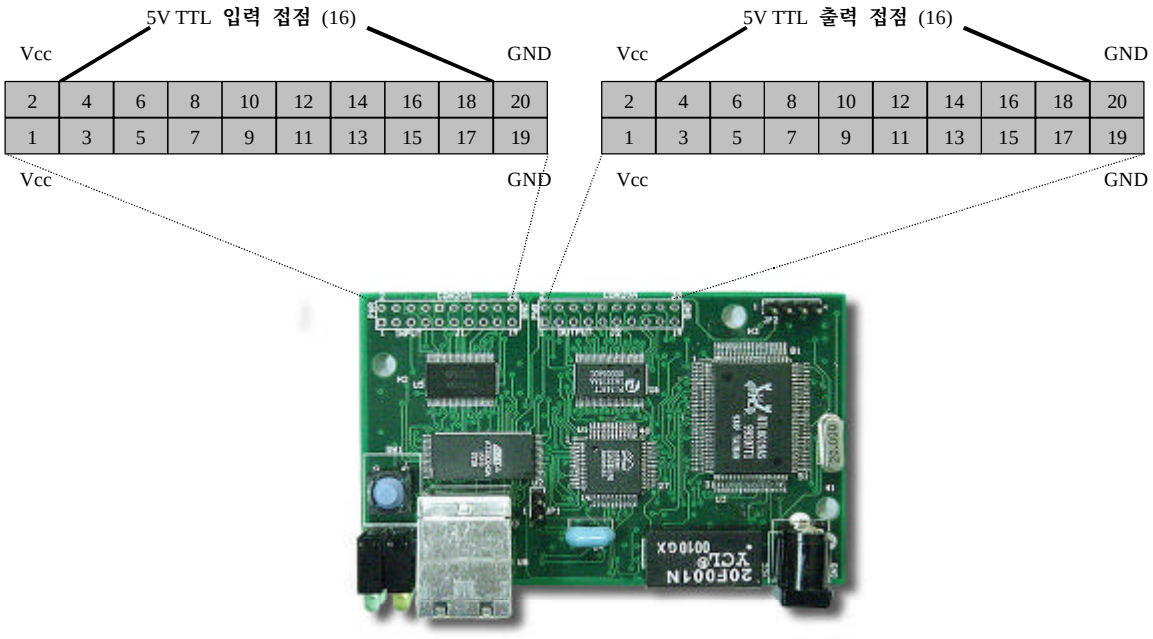


그림 3.4. 입출력 제어 커넥터 핀 할당

3.5 플래시 메모리

사용자가 하부장치를 제어하기 위하여 작성한 웹 페이지 리소스 및 콘트롤러 고유의 물리적 주소 등은 HelloDevice 하드웨어 보드상의 플래시메모리에 저장된다..

- 4 Mbit (512 Kbyte)
- 256 byte 단위의 2048 개의 섹터로 구성됨

3.6 펌웨어

마이크로프로세서의 프로그램 메모리에는 인터넷과의 접속을 위한 개방형 통신환경 (OSI: Open System Interconnection)에 부합하는 TCP/IP 스택 및 하부장치와의 접속을 위한 디지털 입출력 인터페이스 등이 펌웨어의 형태로 저장되어있다.

계층		HelloDevice 지원				
7	**Application**	**HTTP**	**디지털 입출력 제어**	**시스템 설정**	**BOOTP**	
6	**Presentation**					
5	**Session**					
4	**Transport**					
3	**Network**	**TCP**		**UDP**		
2	**Data link**	**IP / ICMP**				**ARP**
1	**Physical layer**	**이더넷 (IEEE802.3)**				

표 3.2. OSI 7 계층과 HelloDevice 1100 펌웨어

4. 메인보드 설치

다음과 같은 순서에 의하여 HelloDevice 1100 을 초기 설치할 수 있다.

(1) HelloDevice 유틸리티 소프트웨어 설치
(2) HelloDevice 전원 연결 및 이더넷 케이블 연결
(3) HelloDevice IP 주소 설정
(4) HelloDevice 네트워크 연결 여부 확인

4.1 유틸리티 소프트웨어 설치

네트워크에 연결되어 있는 PC 에서 HelloDevice CD-ROM 에 있는 `setup1x00.exe` 를 실행한다. `Setup1x00.exe` 는 마이크로소프트 윈도우 95/98, NT 및 윈도우 2000 에서 동작한다. setup type 화면에서 [**Complete**]을 선택하고 [**Next**] 버튼을 누른다.

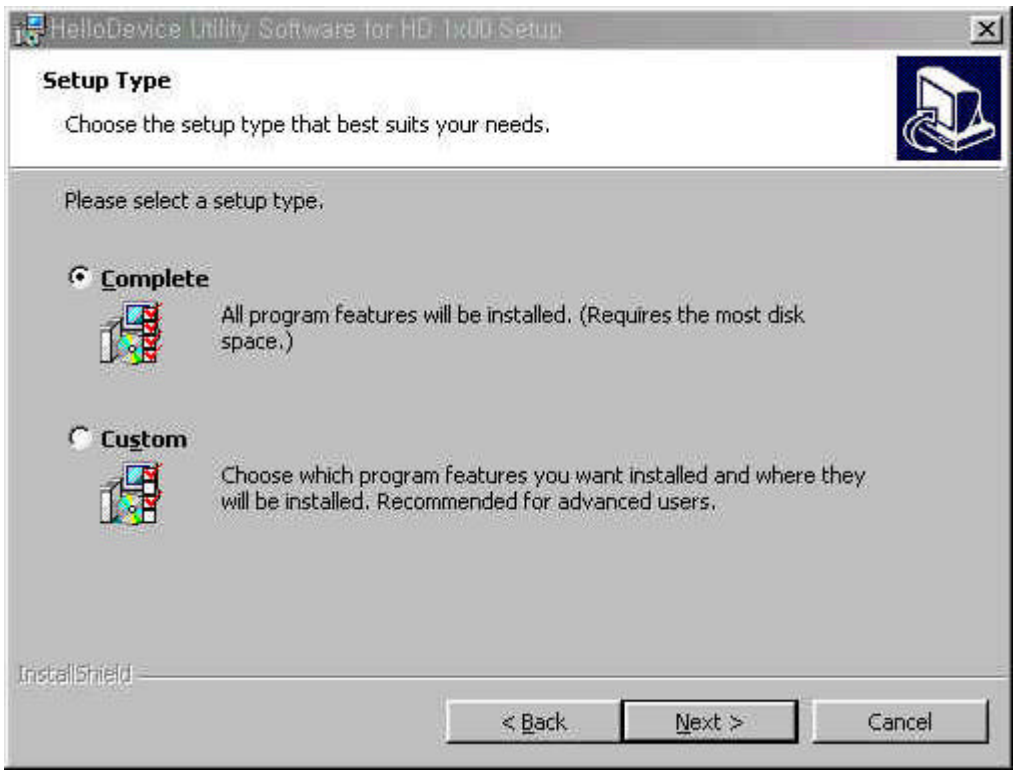


그림 4.1 HelloDevice 유틸리티 소프트웨어 설치 프로그램 초기 화면

설치 프로그램은 자동으로 **c:\Program Files\HelloDevice utility**에 설치된다.

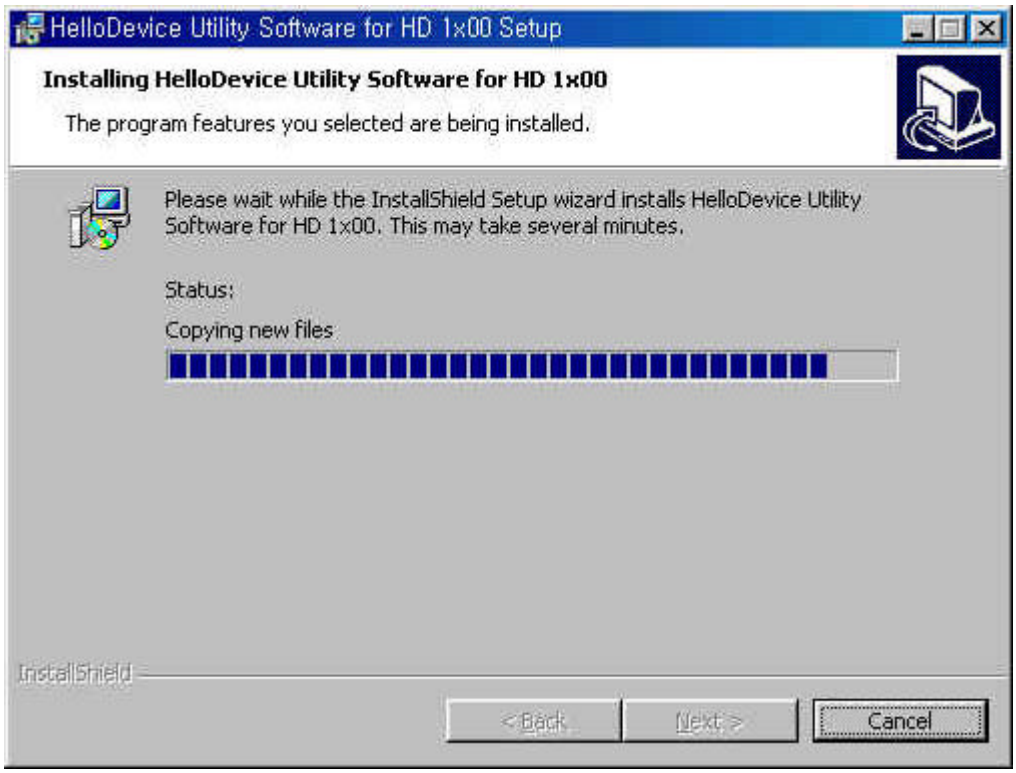


그림 4.2 HelloDevice 유틸리티 소프트웨어 설치 진행 화면

설치가 종료되면 바탕 화면 및 프로그램 메뉴에 HelloDevice 유틸리티 소프트웨어 바로가기 아이콘이 생성되며, 이를 실행하여 다음과 같은 HelloDevice 유틸리티 초기 화면이 나타나면 설치는 성공적으로 완료된다.

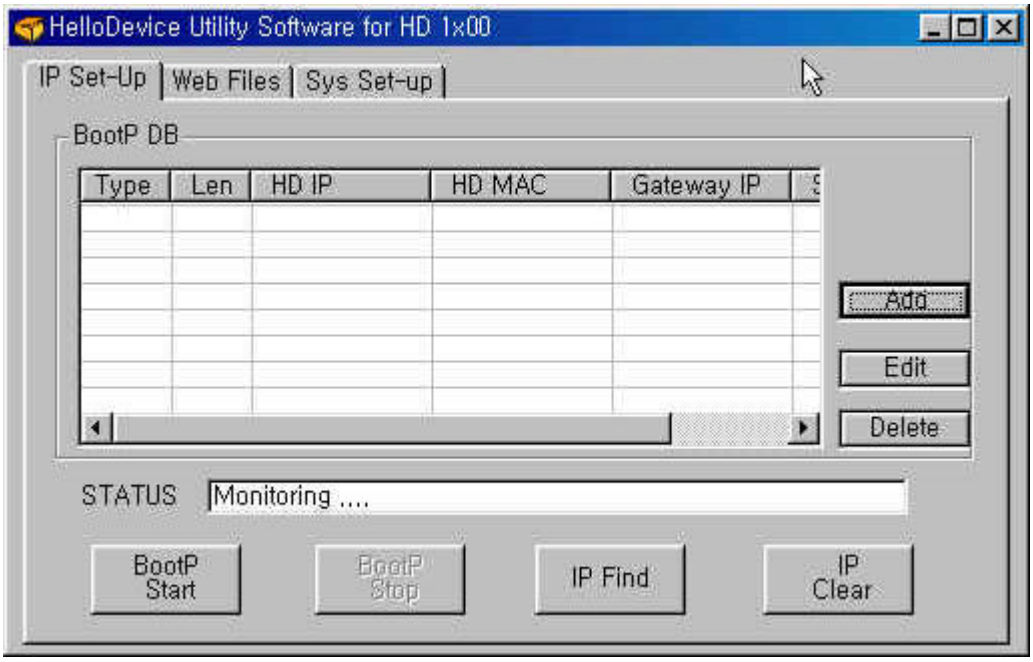


그림 4.3 HelloDevice 유틸리티 소프트웨어 실행 초기 화면

4.2 하드웨어 연결

(1) 5V 전원 어댑터를 HelloDevice 에 연결한다.

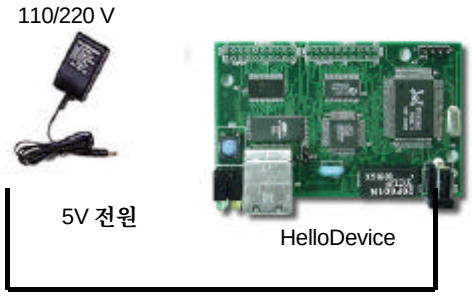


그림 4.4 HelloDevice 와 5V 전원 연결

(2) RJ45 커넥터를 통하여 HelloDevice 와 이더넷 허브를 연결한다.
(HelloDevice 의 RJ45 커넥터 사양은 3 장의 메인보드 사양 부분 참조)

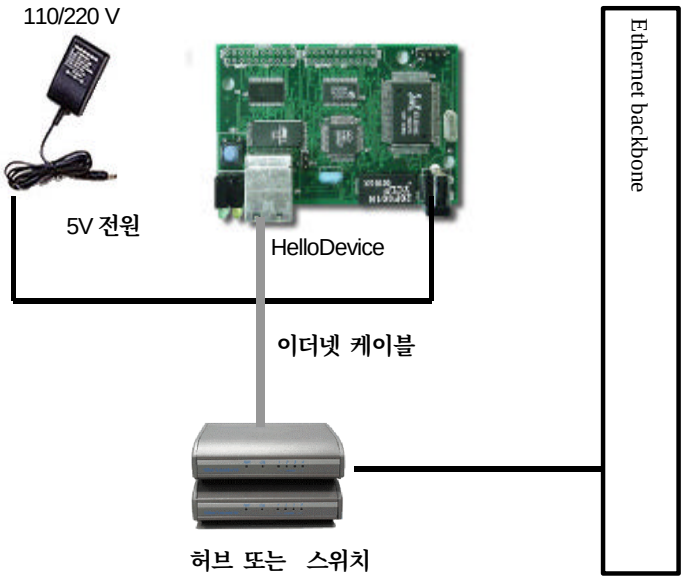


그림 4.5 HelloDevice 와 이더넷 케이블 연결

(3) HelloDevice 의 상태 LED 중 Tx LED 가 1 초마다 주기적으로 점등/점멸되는지 확인한다.
(그림 3.1 의 [상태 LED] 참조)

4.3 네트워크 환경 설정

HelloDevice 가 네트워크 상에서 서버로 동작하기 위해서는, HelloDevice 에 IP 주소를 설정해야 한다. 본 절에서는, IP 주소의 설정 및 변경에 대해 설명하도록 한다.

4.3.1 IP 주소 설정

HelloDevice 에서 IP 주소는 HelloDevice 유틸리티 소프트웨어로 설정할 수 있으며, 이는 인터넷 규약 표준인 RFC-951, RFC-1542 에 정의되어 있는 BOOTP (BOOTstrap Protocol) 에 의해서 수행된다. 유틸리티 소프트웨어는 BOOTP 서버 기능을 지원하며 이 BOOTP 서버는 HelloDevice 의 BOOTP 요청에 대하여 응답하는 역할을 한다.

HelloDevice 의 IP 주소는 공장출하시 0.0.0.0 으로 설정되어 있으므로, 최초에 전원이 연결되면, BOOTP 가 IP 주소의 할당을 요구하게 되므로, HelloDevice 상의 TxLED 가 주기적으로 깜박이게 된다. (그림 3.1 의 [상태 LED] 참조)

HelloDevice 의 IP 주소 관리 및 설정은 IP 주소 데이터베이스에 의하여 이루어지므로, 사용자는 적절히 할당된 HelloDevice 의 MAC[2]-IP 주소 레코드를 IP 주소 데이터베이스에 입력하여야 한다. HelloDevice 의 IP 주소는 사용자의 네트워크 환경에 따라 사용가능한 IP 주소로 할당 되어야 하므로, 사용자는 네트워크 관리자에게 문의하여 사용가능한 IP 주소를 할당받아 사용하도록 한다.

이제 HelloDevice 유틸리티 소프트웨어를 통해 IP 주소를 설정해 보도록 한다.

[2] MAC 주소는, 이더넷 콘트롤러를 제작한 회사 고유의 코드 및 그 회사에서 부여한 제품 코드로 구성되어 있으며, 총 6 byte 로 구성된다. HelloDevice 의 MAC 주소는 세나테크놀로지 고유의 MAC 주소인 00-01-95 와 제품 고유의 코드인 xx-xx-xx 로 구성되어 있다.
예) 00-01-95-01-aa-08, 00-01-95-01-02-01

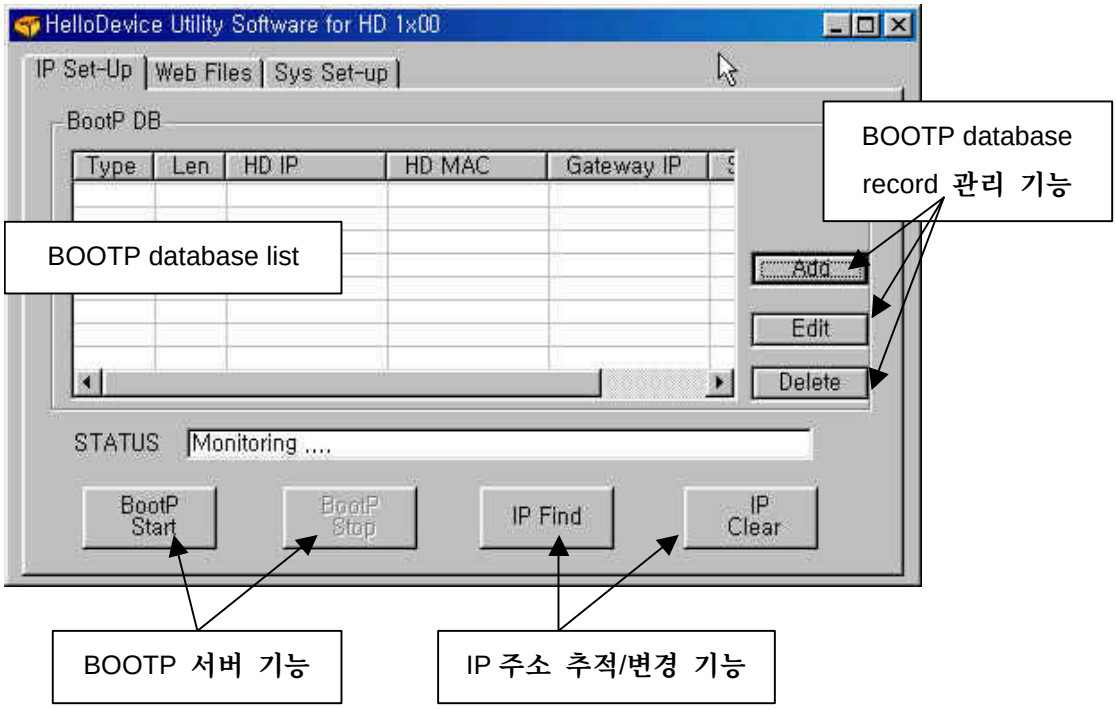


그림 4.6 HelloDevice 유틸리티 소프트웨어의 IP 주소 설정 화면

HelloDevice 유틸리티 소프트웨어를 이용한 IP 주소 설정

(1) 사용자 PC 바탕 화면의 HelloDevice 유틸리티 바로가기를 실행하고, [IP Set-up] 메뉴 탭을 선택하여 [IP Set-up] 창으로 이동한다.

여기에서는 BOOTP 서버 기능을 이용한 IP 주소의 설정([BootP Start], [BootP Stop]), IP 주소의 추적([IP Find]) 및 변경([IP Clear])이 이루어진다.

(2) [Add] 버튼을 눌러서, BOOTP 데이터베이스 레코드를 입력한다.

HelloDevice 고유의 MAC 주소와 IP 주소를 입력한다. H/W address type 및 H/W address length 항목은 HelloDevice 가 이더넷으로 인터넷에 연결되므로, 항상 1 및 6 으로 고정되어 있다. MAC 주소는 HelloDevice 보드의 이더넷 IC 윗면의 스티커 라벨에 기재되어 있다. 그림 4.7 에서는, 라벨에 적힌 MAC 주소가 00:01:95:04:02:03 이고, HelloDevice 에 할당된 IP 주소가 192.168.1.15 일 경우의 입력 예를 나타낸 것이다. 이때, HelloDevice 의 BOOTP 기능은 브로드캐스트 메시지(Broadcast message)의 형태로 동작하므로, IP 라우팅을 위한 게이트웨이 주소 및 서브넷 마스크 값은 입력하지 않아도 무방하다.

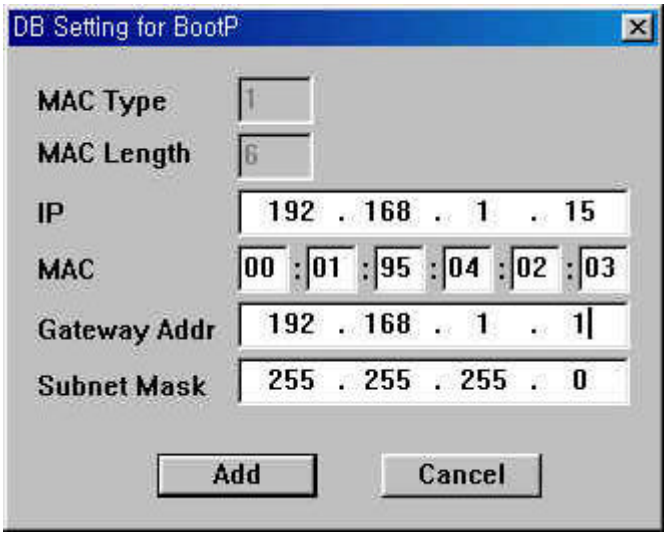


그림 4.7 BOOTP 데이터베이스 레코드 입력 창

(3) [Add] 버튼을 눌러서 데이터베이스 편집을 마친다.

이로써, 과정 (2)에서 입력한 레코드가 IP 데이터베이스에 추가되었다.

(4) [BootP start] 버튼을 누름으로써, BOOTP 서버 기능을 시작한다.

이제 HelloDevice 유틸리티는 BOOTP 서버로서 동작하게 된다. [Status] 바의 상태가 "Monitoring"에서 "Listening BOOTP request"로 바뀐 것을 볼 수 있다.

만약 BootP DB List 에 존재하지 않는 HelloDevice 의 BootP 요구가 있으면 HelloDevice 유틸리티 소프트웨어는 "DB Setting for BootP" 창을 실행하여 사용자의 입력을 받아 BootP DB 에 입력하게 된다.

(5) HelloDevice 의 TX LED 의 상태를 확인한다.

HelloDevice 의 IP 할당 요구에 대해, HelloDevice 유틸리티 소프트웨어는 IP 주소를 할당하기 위한 응답을 전송하게 된다. 이때, [Status] 바는 "BootP reply sent... [192.168. 1. 15]"라는 메시지를 표시한다. HelloDevice 의 TX LED 가 주기적으로 점등/점멸을 반복할 경우는, IP 주소 할당을 요구하는 메시지를 계속 전송하는 것이다. 더 이상 TX LED 가 점등/점멸하지 않을 때까지 기다린 후 [BOOTP Stop] 버튼을 눌러서 BOOTP 서버 기능을 멈추도록 한다.

(6) ping 프로그램을 이용하여, HelloDevice IP 주소 설정을 확인한다.

ping 을 Command prompt 상태에서 실행한다. 정상적으로 설정되었으면, 다음과 같이 ping 에 대한 응답이 되돌아 온다.

```
>> ping 192.168.1.15
>> Pinging 192.168.1.15 with 32 bytes of data:
        Reply from 192.168.1.15: bytes=32 time=10ms TTL=251
```

```
Reply from 192.168.1.15: bytes=32 time<10ms TTL=251
Reply from 192.168.1.15: bytes=32 time=10ms TTL=251
```

만약 위의 응답이 표시되지 않는다면, IP 주소가 올바로 설정되지 않은 것이므로, 과정 (4), (5), (6)을 반복한다.

(7) [IP Find] 기능을 이용하여, HelloDevice IP 주소 설정을 확인한다.

[IP Find] 버튼을 누르면, 그림 4.8 과 같은 창이 나타난다. 설정한 HelloDevice 의 MAC 주소를 입력하고, [Find] 버튼을 누르면, "Found IP" 항목에 현재 설정되어 있는 IP 주소가 나타난다.

그림 4.8 [Find IP] 기능을 이용한 IP 주소 확인

(8) HelloDevice 웹 서버 기능이 동작하는지 여부를 확인한다.

출하될 때, HelloDevice 에는 기본적으로 404.html 파일이 플래쉬 메모리에 내장되게 된다. 웹 서버 기능을 확인하기 위해, 웹 브라우저를 실행하여 설정된 IP 주소로 접속한다. 그림 4.9 의 웹 페이지가 나타난다면 웹 서버로서의 동작이 확인된 것이다.

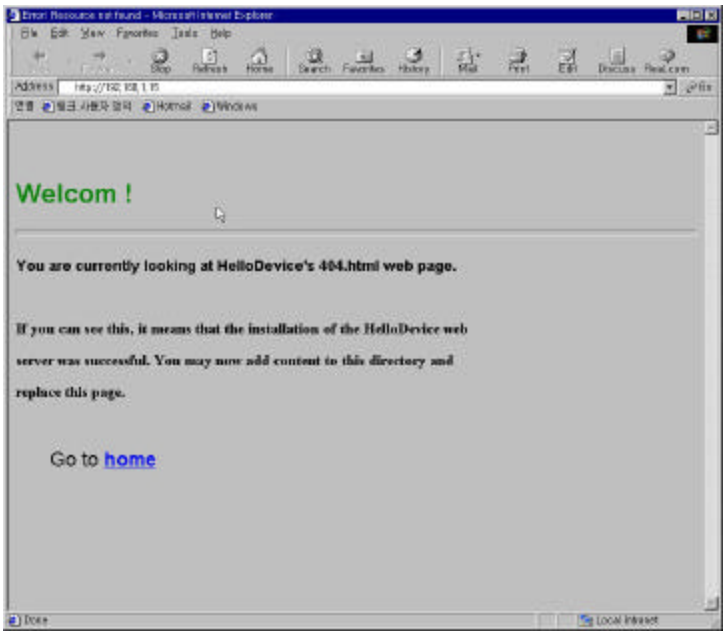


그림 4.9 HelloDevice 기본 설정 화면

4.3.2 IP 주소 변경

이미 설정된 IP 주소를 변경하기 위해서는, 먼저 HelloDevice 의 IP 주소를 0.0.0.0 으로 초기화하고 다시 4.3.1 절의 IP 주소 설정 과정을 반복하여야 한다. 이때, 사용자의 네트워크 관리자와 상의하여 적절한 IP 주소를 재할당 받도록 한다.
여기서는, 설정한 IP 주소인 192.168.1.15 을 192.168.1.18 로 수정하고자 한다고 가정하고, IP 주소 변경 과정을 밟아 나가도록 한다.

(1) IP 주소의 초기화

IP 주소 clear 명령을 실행하기 전에, 사용자 PC 에 저장되어 있는 HelloDevice 의 기존 IP/MAC 주소 데이터를 제거해주어야 [IP clear] 명령이 올바르게 동작되게 된다. 이 데이터는 사용자 PC 의 ARP 캐쉬에 저장되어 있으며, 다음과 같은 명령으로 사용자 PC 의 ARP 캐쉬 저장값을 확인할 수 있다. 본 예는, 사용자 PC 의 IP 주소가 192.168.1.100 인 경우의 응답을 나타낸다.

```
>>arp -a
Interface: 192.168.1.100 on Interface 2
  Internet Address      Physical Address      Type
  192.168.1.15          00-01-95-04-02-03     dynamic
  192.168.1.23          01-a0-11-34-11-0d     dynamic
```

현재 HelloDevice 의 IP 주소 설정치에 해당하는 항목을 ARP 캐쉬에서 제거하도록 한다.

```
>>arp –d 192.168.1.15
```

이제 IP 주소를 초기화할 준비가 되었다.

[IP Setup] 창으로 이동한 후 [IP Clear] 버튼을 선택하면, IP 주소 변경을 위한 창이 나타난다. 사용자가 현재 설정한 IP 주소인 192.168.1.15 를 0.0.0.0 으로 초기화하고자 하면, 다음과 같이 현재의 설정된 IP 주소를 입력하고, [OK] 버튼을 누른다. 그러면, 현재 IP 주소가 192.168.1.15 인 HelloDevice IP 주소가 0.0.0.0 으로 초기화되게 된다.

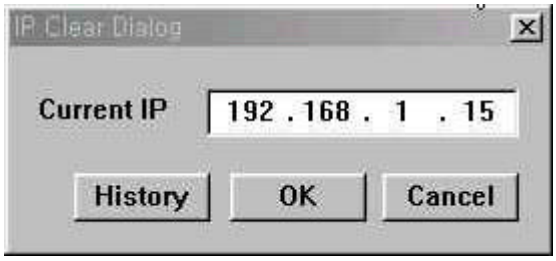


그림 4.10 IP 주소 Clear 창

(2) IP 주소의 초기화 확인

HelloDevice 의 TX LED 가 깜박거리는지 확인한다. IP 주소가 초기화되었다면, IP 주소 할당을 요구하는 메시지를 네트워크를 통해 전송하므로, HelloDevice 의 TX LED 가 계속적으로 점등/점멸을 반복하게 될 것이다.

(3) IP 주소의 재설정

IP 주소를 192.168.1.18 로 재설정하기 위해, IP 주소 데이터베이스의 [Edit] 기능을 통해 IP 주소 데이터베이스의 내용을 편집/수정하고, 4.3.1 절의 IP 주소 설정 과정을 반복하도록 한다.

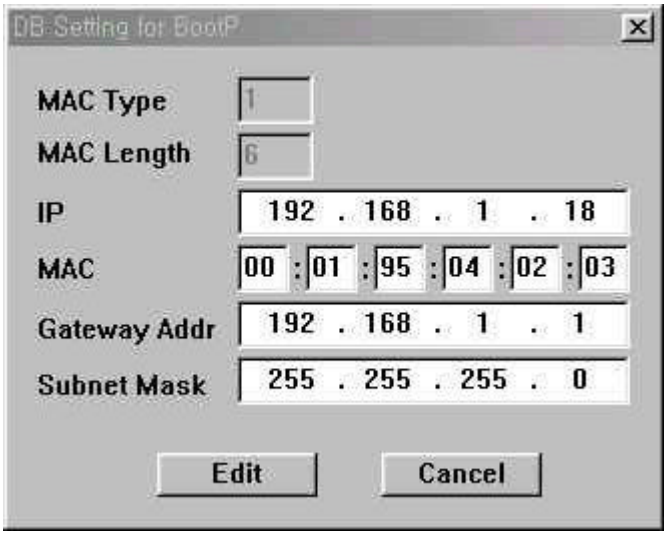


그림 4.11 IP 데이터베이스의 [Edit] 기능을 이용한 IP 주소 수정

5. 맛보기

본 장에서는 예제를 통하여 HelloDevice 웹 서버에 홈페이지를 구축해 보고, HelloDevice 와 입출력 데모 보드를 통해 입출력 접점을 제어해 보도록 한다. HelloDevice Starter's Kit 에서는 사용자의 응용에 도움을 주기 위해, 다음과 같은 예제 파일들을 제공한다.

(1) 첫번째 웹 서버 데모 예제
(2) 웹을 통한 입출력 제어 데모 예제
(3) 입출력 데모 예제의 자바 애플릿 소스 코드
(4) C 프로그램을 이용한 입출력 데모 예제

5.1.절에서는, 예제 (1)을 이용하여, 웹 서버를 최초로 동작시켜 보며, 5.2.절에서는, 예제 (2)를 이용하여 웹을 통해 데모 보드의 입출력 장치들을 제어해 보도록 하자. 예제 (3), (4)는 6 장 일반 기기 대상 입출력 제어 기능 구현에서 설명하도록 한다.

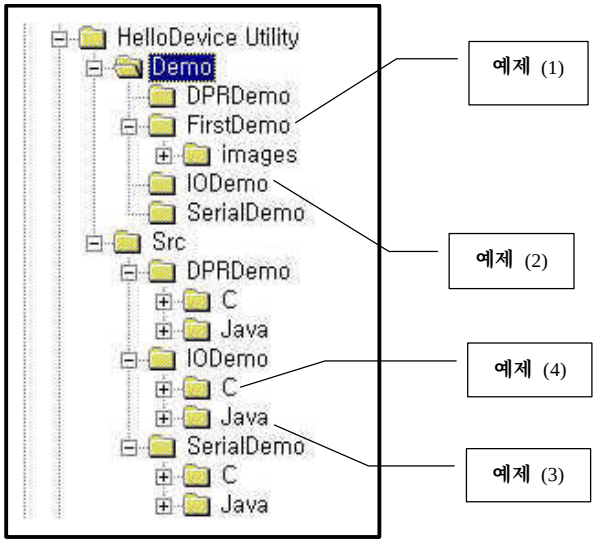


그림 5.1. HelloDevice 예제 파일들의 위치

5.1 홈페이지 구축

HelloDevice 가 웹 서버 기능을 수행하기 위해서는, HTML 파일, 이미지 파일, 자바 클래스 파일 등 웹과 관련된 파일들이 HelloDevice 플래쉬 메모리에 저장되어야 한다. 최대 저장 가능한 파일 개수는 최대 256 개이며 최대 크기는 500 Kbyte 이다.

HelloDevice 는 일반 운영체제에서와 같은 파일시스템을 사용하지 않으므로, 웹 파일들은 "파일 index table"의 형태로 관리된다. 웹파일들을 HelloDevice 플래쉬 메모리에 저장할 때는, 이 들을 하나로 통합하여 하나의 이미지 파일로 ***Build*** 한 후 ***Upload*** 하여 저장하게 된다. 이 파일은 파일 index table 을 포함하고 있으며, HelloDevice 웹 서버만이 인식할 수 있는 형태의 것이다. HelloDevice 유틸리티 소프트웨어에는 이러한 이미지 파일의 Build 및 Upload 기능이 포함되어 있다.

HelloDevice 를 웹 서버로 동작하게 만들기 위해서는, 다음과 같은 작업을 수행하여야 한다.

(1) 웹 페이지 제작 도구를 이용하여 웹 페이지를 제작한다.
(2) HelloDevice 유틸리티 소프트웨어를 사용하여, 제작된 웹 페이지를 HelloDevice 에 업로드할 수 있는 형태로 변환한다.
(3) 변환된 파일을 HelloDevice 에 업로드한다.
(4) 웹 브라우져를 이용하여, 웹 페이지에 접속한다.

이번 예제에서는, "FirstDemo" 폴더의 웹 파일들을 이용하여, 웹 서버 홈페이지를 구성해 보도록 하자. 이 예제 파일은 index.html 을 홈 페이지 파일로 하여, HelloDevice 블록도, 자바 애플릿을 이용한 큐브 애니메이션 데모로 이루어져 있다.

(1) [Web files] 탭을 선택하여 웹 파일 관리 도구가 있는 화면으로 이동한다.

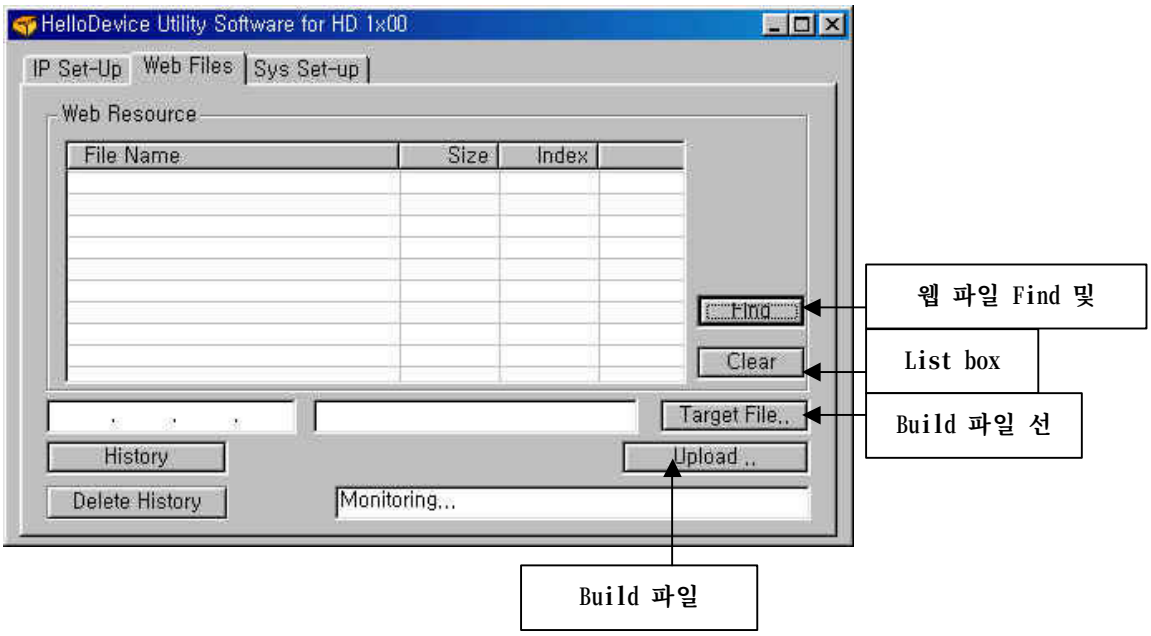


그림 5.2. HelloDevice 유틸리티 소프트웨어 웹 파일 도구 화면

(2) [Find] 버튼을 눌러서 Build 할 웹 파일들을 찾아보도록 하자.

예제 파일이 있는 폴더로 이동하여 최상위 폴더인 “FirstDemo” 폴더를 선택하고, [확인] 버튼을 누른다.

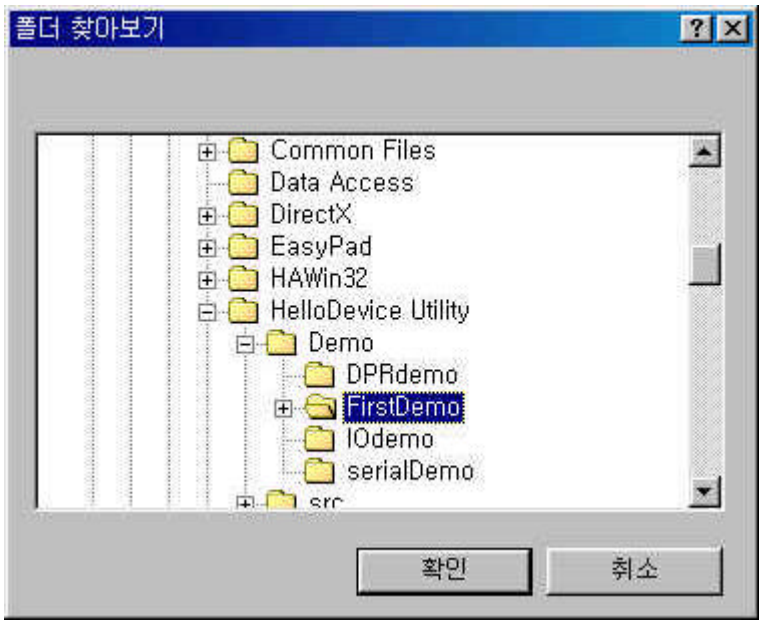


그림 5.3. 웹 서버 데모 예제 파일 찾기

(3) Build 할 파일 이름을 선택한다.

사용자는 저장할 Build 이미지 파일 이름을 입력하고, [저장] 버튼을 누른다. 기본적으로, Build 된 HelloDevice 웹 파일은 *.hd 의 포맷을 갖는다. 현재 화면이 종료되고, “Build complete”이란 대화상자가 나타나게 된다.

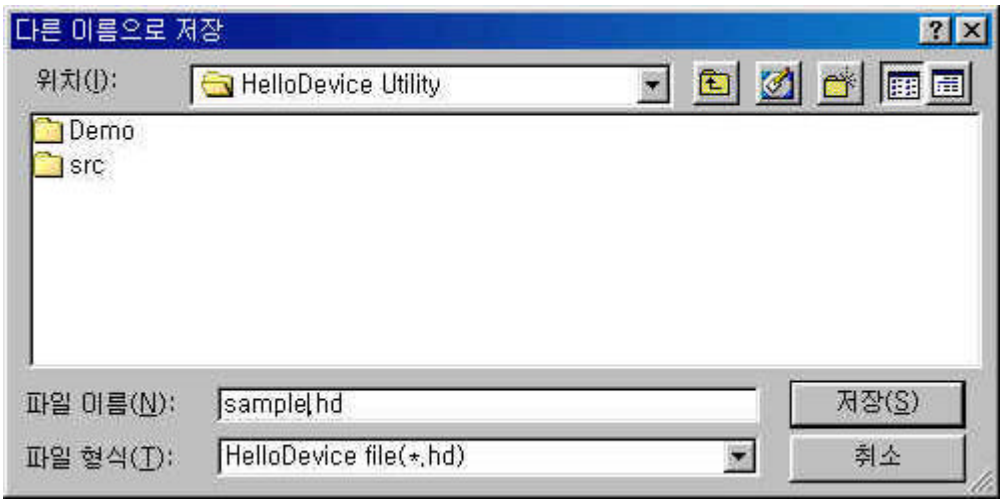


그림 5.4. 웹 예제 Build 파일 저장

(4) Build 된 파일을 업로드할 HelloDevice 의 IP 주소를 입력한다.

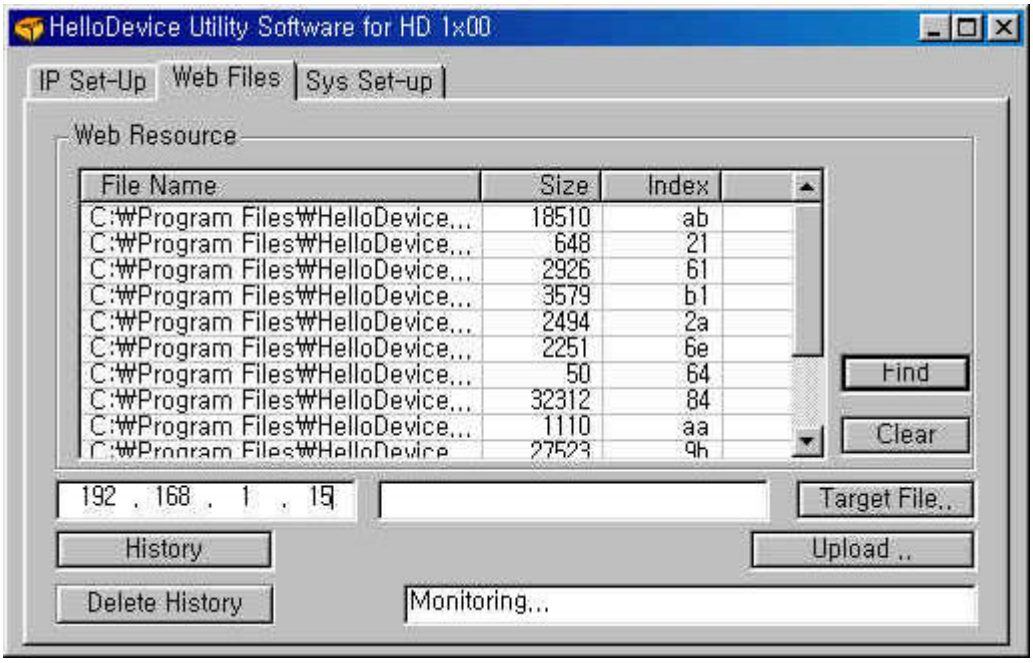


그림 5.5. Build 파일을 업로드할 IP 주소 입력

(5) [Target file..] 버튼을 눌러 업로드할 Build 파일을 선택한다.

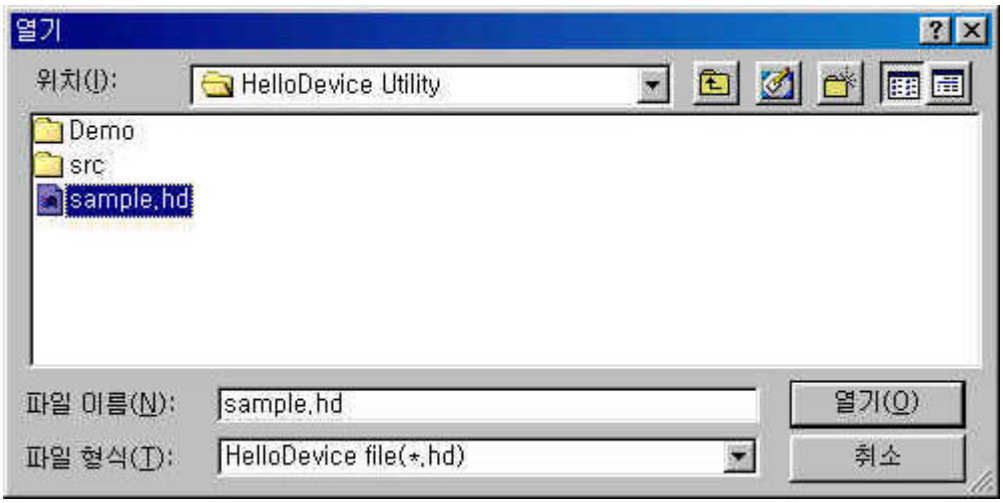


그림 5.6. 업로드 Build 파일 선택 화면

(6) [Upload] 버튼을 눌러 Build 된 이미지 파일을 HelloDevice 로 업로드한다.

이 과정은 Progress bar 를 통해 나타날 것이고, 완료되면 "Flash download completed!!"의 대화 상자가 나타날 것이다.

(7) 웹브라우저로 URL 을 http://192.168.1.15/index.html로 입력하고, 연결을 시도해 본다.

그림 5.7 과 같은 웹 페이지가 나타날 것이다. 만약 그림과 같은 웹 페이지가 나타나지 않는다면

웹 서버가 정상적으로 동작하지 않는 것이므로, 웹서버의 IP 주소 설정 및 파일 업로드 등을 다시 한번 확인한다.

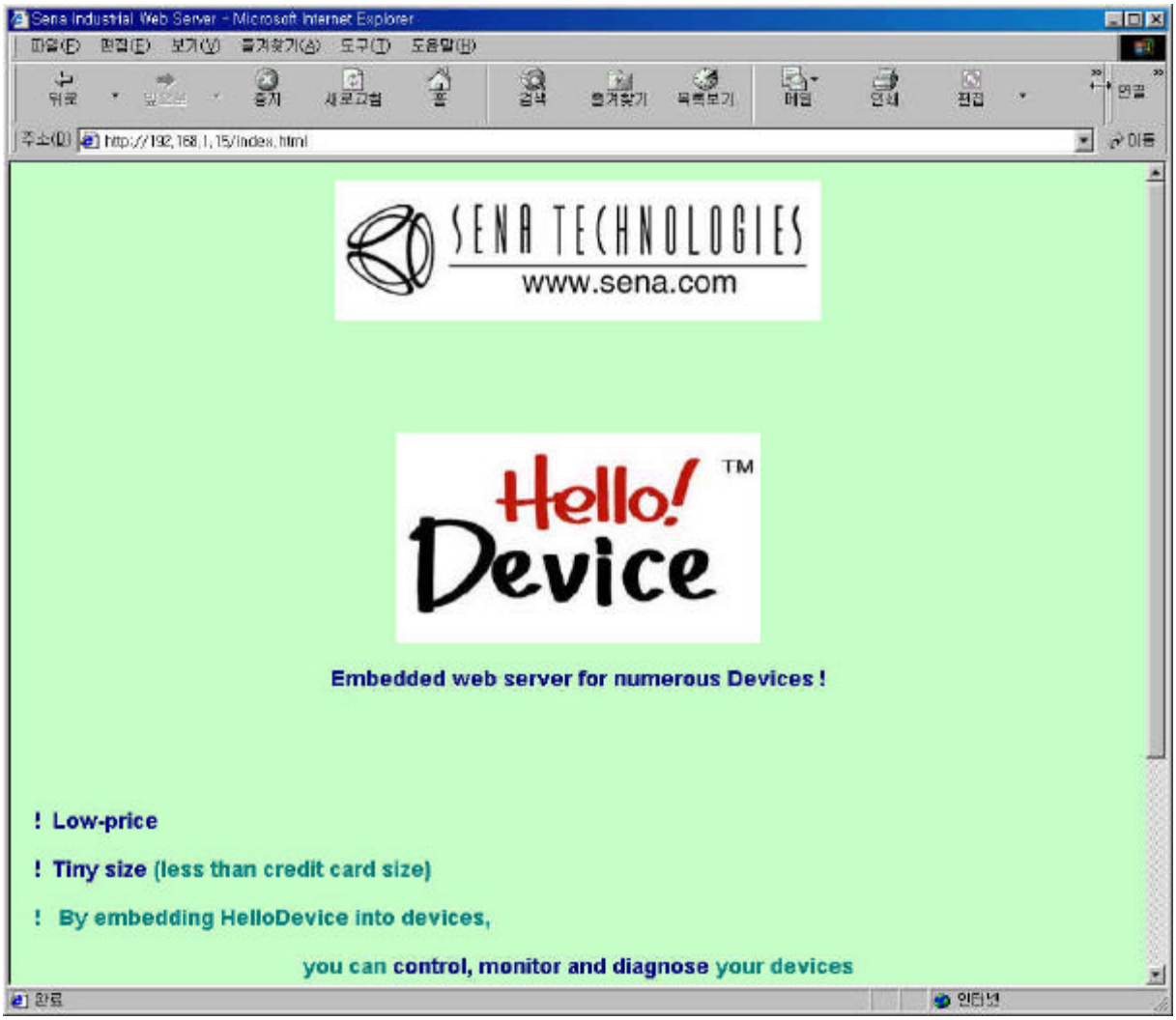


그림 5.7. HelloDevice, “FirstDemo” 웹 페이지 화면

사용자는 나름대로의 웹 페이지를 제작한 후, 본 절에서 설명한 과정을 그대로 따르면, HelloDevice 를 사용자 고유의 웹 페이지를 내장한 웹 서버로서 동작시킬 수 있다.

5.2 데모보드를 이용한 입출력 제어 기능 데모

HelloDevice 1100 Starter's Kit 에서는, HelloDevice 의 디지털 입출력 제어 기능 데모를 위해 데모 보드 및 예제 프로그램을 제공한다. 이들을 이용하여 HelloDevice 입출력 제어를 실험해보도록 하자.

5.2.1 데모보드 연결

(1) HelloDevice 의 전원을 끄고, 이더넷 케이블을 제거한다.
(2) HelloDevice 를 디지털 입출력 커넥터를 통해 데모 보드에 삽입한다.

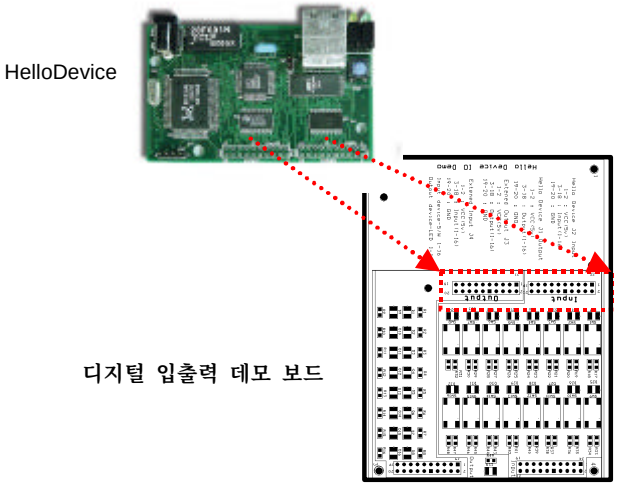


그림 5.8. HelloDevice 를 데모 보드에 삽입하는 방법

(3) HelloDevice 에 다시 전원을 연결하고, 이더넷 케이블을 연결한다.

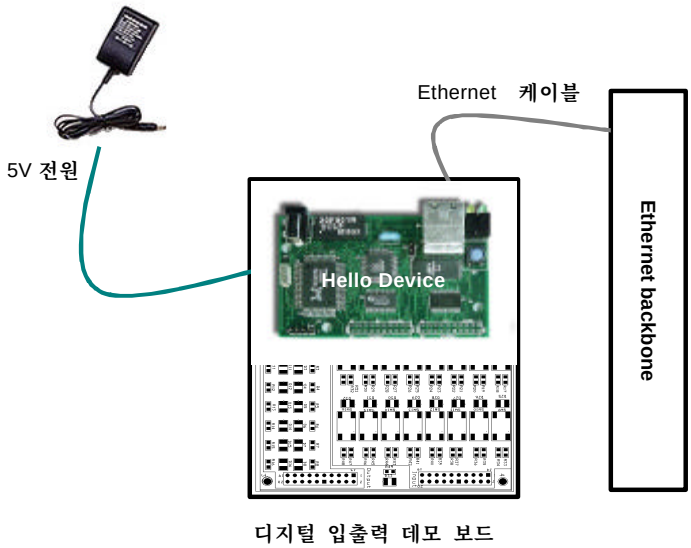


그림 5.9. HelloDevice 디지털 입출력 데모를 위한 하드웨어 설치

5.2.2 웹 데모

웹 상에서, 디지털 입출력 제어를 위해서는 HelloDevice 유틸리티 소프트웨어의 "IODemo" 폴더에 있는 예제 웹 파일을 이용한다. (그림 5.1. 예제 파일들의 위치 참조) 디지털 입출력 데모에 사용되는 웹 파일들은 index.html, io.jar, 404.html 이며, 이 예제에서는, HelloDevice 와 연결된 데모 보드의 입출력을 웹으로 제어하는 것을 보여준다.

(1) "IODemo" 폴더의 index.html 파일을 Notepad 등의 편집기에서 열어 IP 주소를 수정한다.
index.html 파일내의 자바 애플릿과 관련된 IP 주소 파라미터에 현재 설정되어 있는 HelloDevice 의 IP 주소를 기입한다. 나머지 파라미터는 수정하지 않고, 그대로 둔 상태에서 파일을 저장하도록 한다.

```
<HTML>
<HEAD>
<TITLE>IOdemo</TITLE>
</HEAD>
<BODY>
<H1>IOdemo</H1>
<APPLET CODE=IOdemo.class ARCHIVE=io.jar WIDTH=250 HEIGHT=300>
<param name=host value="192.168.1.15">
<param name=port value=6001>
<param name=polling value=1>
</APPLET>
</BODY>
</HTML>
```

그림 5.10. 디지털 입출력 제어용 index.html 파일의 수정

Note:

입출력 제어용으로 제작된 HelloDevice 자바 애플릿 예제는 다음과 같은 3 개의 파라미터를 갖는다.

IP 주소 : 사용자가 설정한 HelloDevice 의 IP 주소
TCP 통신 포트 : 항상 6001 로 설정한다.
Polling 주기 : 100 ms 단위, 사용자의 응용에 따라 적절한 Polling 주기 설정
예) polling value =1 은 100 ms 마다 입출력 상태를 Read 함을 의미한다.

(2) "IODemo" 폴더의 웹 파일들을 [Build] 및 [Upload]하도록 한다.
5.1 절의 [홈페이지 구축]에서 설명된 과정을 밟아서 HelloDevice 에 웹 페이지를 업로드하도록 한다. 단, 웹 파일이 위치한 폴더를 5.1 절의 "FirstDemo" 가 아닌 "IODemo"로 하여 과정들을 다시

수행하도록 한다.

(3) 업로드가 완료되었으면, 웹 브라우저를 실행하여, http://192.168.1.15/index.html로 접속하면, 다음 그림과 같은 I/O 데모 웹페이지가 나타난다. 이 화면은 데모 보드의 스위치 및 LED 와 일대일로 대응되며, 그 대응관계는 그림 5.12 에 나타내었다.

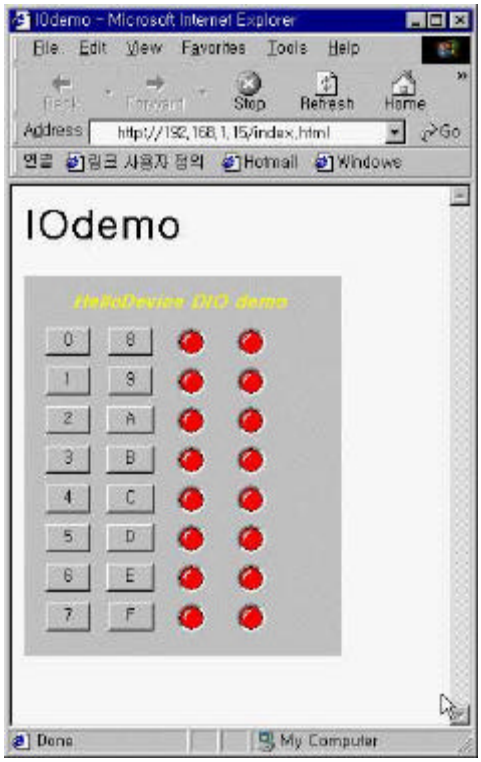


그림 5.11. 디지털 입출력 데모 웹 페이지 화면

(4) 디지털 출력 제어를 확인한다.

웹 페이지 왼쪽에 위치한 버튼들은 디지털 출력 명령을 위한 것이며, 버튼 숫자들은 각각 출력 접점의 번호를 나타낸다. 웹 페이지의 버튼 0 을 누르면, 데모 보드의 LED 0 이 점등될 것이며, 다시 누르면 LED 0 이 점멸될 것이다. 이와 같이 조작하여, 실제 데모 보드에 위치한 LED 의 점등/점멸을 확인한다.

(5) 디지털 입력 감지를 확인한다.

웹 페이지의 오른 쪽에 위치한 LED 는 디지털 입력 감지를 위한 것이며, 데모 보드의 스위치들에 대응된다. 데모 보드에 위치한 스위치를 눌러서, 웹 페이지의 해당되는 LED 색이 회색에서 빨간색으로 바뀌는지 여부를 확인한다.

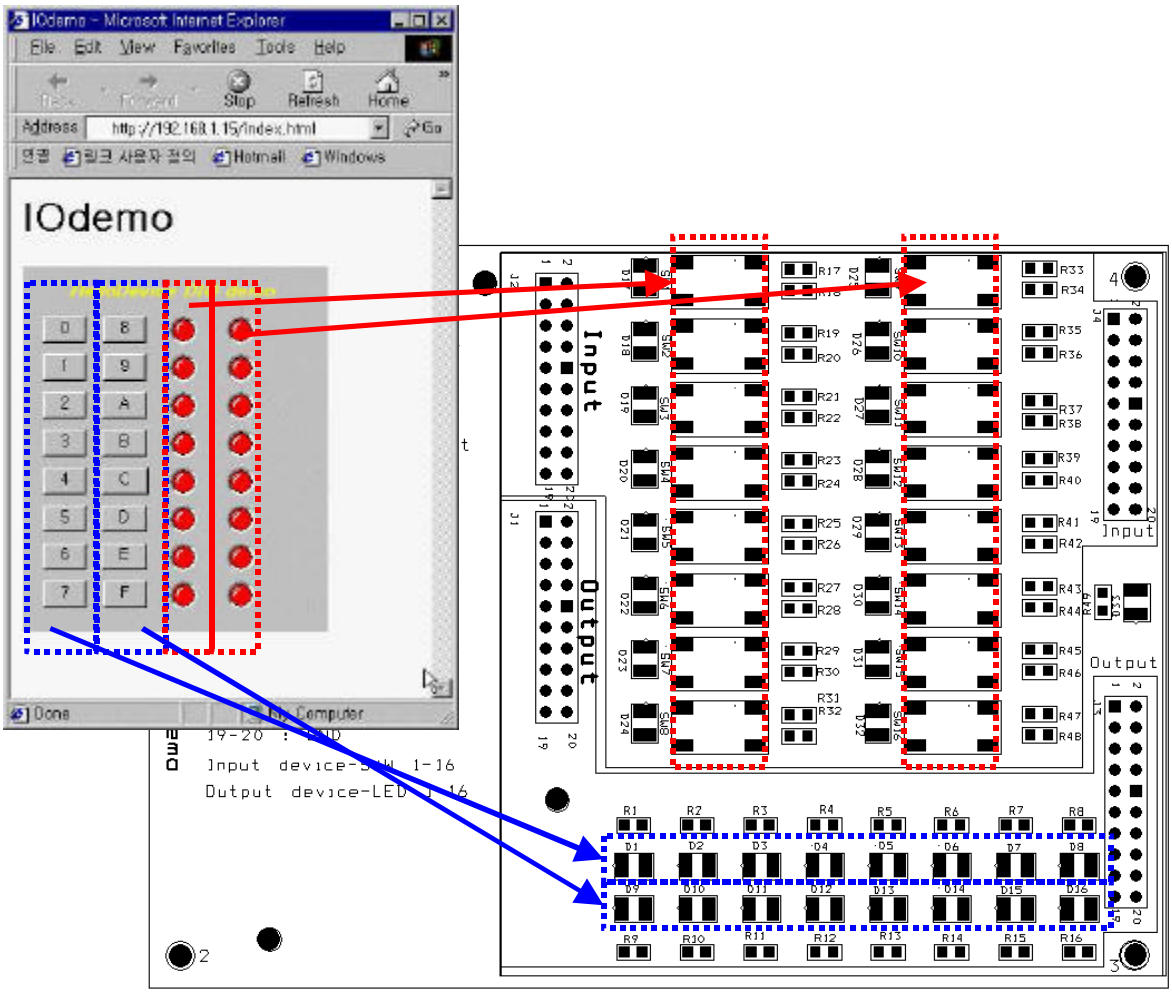


그림 5.12. 디지털 입출력 데모 화면과 데모 보드와의 상관 관계

6. 일반 기기 대상 입출력 제어 기능 구현

사용자 PC 에서 HelloDevice 에 연결된 입출력 장치를 제어하기 위해서는, 사용자의 PC 의 인터넷 소켓 인터페이스 프로그램을 사용하여 HelloDevice 로 TCP/IP 프로토콜의 입출력 제어 명령을 전송하여야 한다. 이때, HelloDevice 는 자신의 입출력 인터페이스의 출력 접점을 HIGH 또는 LOW 로 동작 시키거나, 입력 접점의 값이 HIGH 인지 LOW 인지를 인식하여 네트워크를 통해 그 결과치를 전송하게 된다. 이 결과치는 사용자 PC 의 인터넷 소켓 인터페이스 프로그램에 전송됨으로써, 인터넷을 통한 원격 명령 및 모니터링 기능을 수행한다.

HelloDevice 의 원격 제어/모니터링 구성도 및 이를 위한 명령/응답 경로를 요약하면 그림 6.1 과 같다. 결국 사용자가 HelloDevice 를 이용하여 인터넷을 통한 하드웨어의 제어 및 감시 기능을 구현하기 위해서는

- HelloDevice 통신 프로토콜에 의거한 자바 애플릿 또는 인터넷 소켓 프로그램의 구현
- HelloDevice 입출력 인터페이스에 부합되는 하드웨어의 구현 및 연결

의 두가지 작업을 수행하여야 한다.

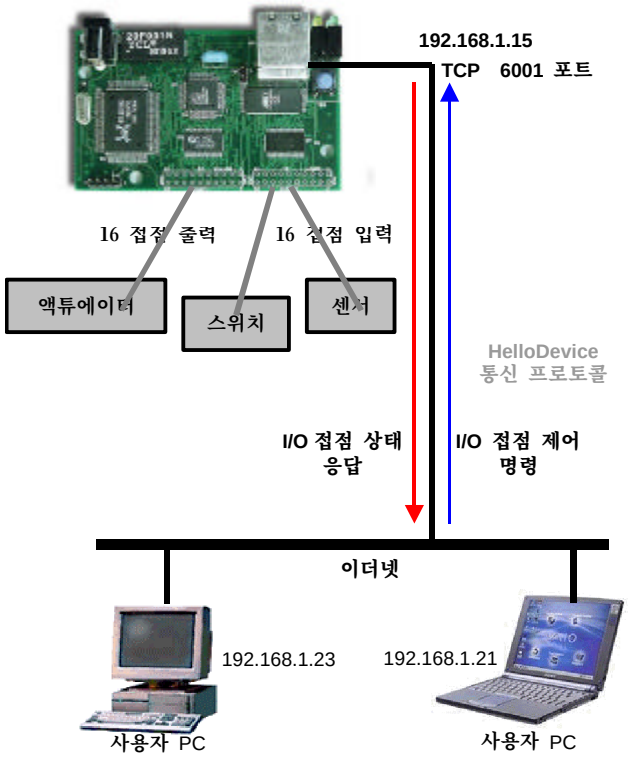


그림 6.1. HelloDevice 의 원격 제어/모니터링

HelloDevice 를 이용한 사용자 어플리케이션의 구현을 위하여, 6.1.절에서는, HelloDevice 통신 프로토콜에 대해 설명하도록 하고, 6.2 절에서는 입출력 데모 보드 및 HelloDevice 입출력 인터페이스에 부합하는 사용자 하드웨어를 구현하는 방법을 설명하도록 한다.

6.1 통신 프로토콜

인터넷 기반 원격 입출력 제어를 위해서는, HelloDevice 와의 인터넷 통신을 위한 인터넷 소켓 인터페이스가 사용자의 PC 에서 구현되어야 한다. 이 소켓 인터페이스는

- C 언어에 기반한 소켓 프로그램
- 웹에서 동작하는 자바 애플릿의 형태

등으로 구현된다.

HelloDevice 와 통신하기 위한 인터넷 소켓 인터페이스는, TCP/IP 프로토콜에 따라 구현되어야 하며, HelloDevice 의 현재 IP 주소 및 TCP 포트 6001 을 목적지로 하여 입출력 제어 명령을 전송하게 된다. HelloDevice 에서는 이를 해독하여, 실제 HelloDevice 와 연결된 입출력 장치들을 제어하게 된다. 소켓 인터페이스와 HelloDevice 사이의 통신 명령 및 응답은 표 6.1 과 같이 요약될 수 있다.

동작 종류	통신 패킷	방향	
		사용자 PC	HelloDevice
입출력 상태 모니터링	입출력 Get 명령	→	
	입출력 Get 명령 응답	←	
출력 제어	출력 Set 명령	→	

표 6.1. HelloDevice 입출력 제어 통신 명령 및 응답

6.1.1 입출력 상태 모니터링

사용자 PC 의 소켓 인터페이스에서, HelloDevice 의 입출력 상태를 읽기 위해 HelloDevice 로 전송하는 명령 및 이에 따른 응답으로 동작된다. 그림 6.2 에 모니터링 명령 및 응답, 그리고 입출력 접점 상태를 해석하기 위한 표를 나타내었다.

종류	Byte 수			
	1	2	3	4
입출력 상태 모니터링 명령	0x75			
입출력 상태 모니터링 응답	입력 접점 상태		출력 접점 상태	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
입력 접점 MSB								입력 접점 LSB								출력 접점 MSB								출력 접점 LSB							
입력 접점 상태																출력 접점 상태															

그림 6.2. HelloDevice 입출력 통신 프로토콜

단, HelloDevice 로부터의 입출력 모니터링 응답은 사용자의 입출력 장치가 HIGH Active 로 동작하느냐, LOW Active 로 동작하느냐에 따라 다르게 나타난다. 현재 HelloDevice 입출력 데모 보드의 입력 스위치는 LOW Active 이고, 출력 LED 는 HIGH Active 로 동작되도록 설계되어 있다. 본 절에서의 입출력 제어 통신 예는 HelloDevice 데모 보드의 경우라고 가정하고, 설명하도록 한다.

예를 들어, 입력 접점 0, 2, 4, 10 만이 HIGH 상태이고, 출력 접점 1, 3, 5 만이 HIGH 상태이면, HelloDevice 는 그림 6.3 에 나타난 바와 같이 16 진수로 FB:EA:00:2A 의 응답을 전송할 것이다. 소켓 인터페이스부에서는 이를 해석하여, 입력 접점 0, 2, 4, 10 만이 HIGH 상태이고, 출력 접점 1, 3, 5 만이 HIGH 상태인 것으로 표시해야 할 것이다.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
1111 | 1011 | 1110 | 1010 = 0xFBEA																0000 | 0000 | 0010 | 1010 = 0x002A															
입력 접점 상태																출력 접점 상태															

그림 6.3. 입출력 상태 해석

Note:

입출력 상태 모니터링 명령의 응답에서, 실제로는 입력 16 점, 출력 16 점의 도합 32 점에 따른 32 bit, 즉, 4 byte 의 데이터 만이 필요하지만, HelloDevice 에서 사용자 PC 로 전송할 때는 항상 16 byte 를 전송한다. 따라서, 사용자는 수신하는 패킷의 첫번째 4 바이트의 내용만 읽어서 사용하여야 한다. HelloDevice 에서 이러한 방식으로 데이터를 전송하는 이유는, 일반적으로 PC 의 LAN 카드에서 수신하는 이더넷 패킷의 최소 바이트 수가 이더넷, IP, TCP, UDP 등 각각의 통신 규약의 헤더를 포함하여 총 64 byte 로 보통 설정되어 있기 때문이다. 보통 LAN 카드에서 설정된 최소 이더넷 패킷 크기보다 작은 패킷은 LAN 카드에서 무시되므로, HelloDevice 에서 사용자 PC 로 패킷을 전송하더라도 사용자의 소켓 인터페이스부로 도달하지 못하게 된다.

따라서, 실제 사용자의 소켓 인터페이스부에서 수신하는 패킷은 다음과 같게 된다.

2 bytes	2 bytes	12 bytes
입력 접점 상태	출력 접점 상태	NULL 데이터

6.1.2 출력 제어 명령

사용자가 HelloDevice 의 출력 접점을 원하는 대로 HIGH/LOW 로 동작시키기 위한 명령이며, 이에 따른 응답은 없다.

종류	Byte 수		
	1	2	3
출력 상태 제어 명령	0x76	출력 접점 상태	

예를 들어, 사용자가 HelloDevice 의 출력 접점 0, 1, 2, 3 을 HIGH 로 나머지는 LOW 로 동작시키고 싶다면, 그 값은 0x000F 로 되며, 그림 6.4 와 같이 16 진수로 76:00:0F 를 HelloDevice 로 전송하면 될 것이다.

출력 제어 명령	출력 접점 MSB								출력 접점 LSB							
8 ~ 0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0x76	0x00								0x0F							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0000 | 0000 | 0000 | 1111 = 0x000F															
출력 접점 상태															

그림 6.4. 출력 제어 명령 예

6.2 하드웨어 구현

본 절에서는, HelloDevice 와 사용자의 입출력 장치와의 인터페이스 구현 예를 보여주기 위해 제작된 입출력 데모 보드에 대해 설명하고, 이를 바탕으로 사용자의 하드웨어 구현 방법에 대해 설명하도록 한다.

6.2.1 HelloDevice 입출력 데모 보드

HelloDevice 입출력 데모 보드는 16 개의 입력 데모용 스위치, 16 개의 입력 표시 LED 및 16 개의 출력 표시 LED 로 구성되어 있다. 입력 데모용 스위치는 LOW Active 로, 출력 표시 LED 는 High Active 로 동작하도록 설계되어 있다.

입력 데모용 스위치	HelloDevice 입력접점	입력표시 LED
CLOSED OPEN	LOW HIGH	ON OFF

HelloDevice 출력접점	출력표시 LED
LOW HIGH	OFF ON

데모 보드의 전원은 20 핀 커넥터를 통해, HelloDevice 로부터 공급 받도록 설계되어 있으며, HelloDevice 의 입출력 인터페이스부 및 입출력 데모 보드의 핀 할당은 그림 3.4 를 참조하도록 한다. 그림 6.6 에는 HelloDevice 입출력 제어용 데모 보드의 회로도를 나타내었다. 사용자는 이 회로도를 참고로 하여, 사용자의 입출력 하드웨어를 설계할 수 있다.

HelloDevice 와 입출력 데모 보드를 연결하기 위해서는, HelloDevice 의 입력부는 J1, HelloDevice 의 출력부는 J2, 두개의 20 핀 커넥터를 이용한다. 또한, HelloDevice 의 J1 과 입출력 데모보드의 J3, HelloDevice 의 J2 와 입출력 데모보드의 J4 는 바이패스 연결되어 있으므로, 사용자는 입력부는 데모보드의 J3, 출력부는 데모보드의 J4 를 통해 사용자 입출력 장치와 연결할 수 있다. 이와 같이 데모보드의 J3, J4 를 연결하여 사용하면 이 Starter Kit 에 포함된 입출력 데모 보드를 사용자 입출력 장치의 디버깅용으로 사용할 수 있다.

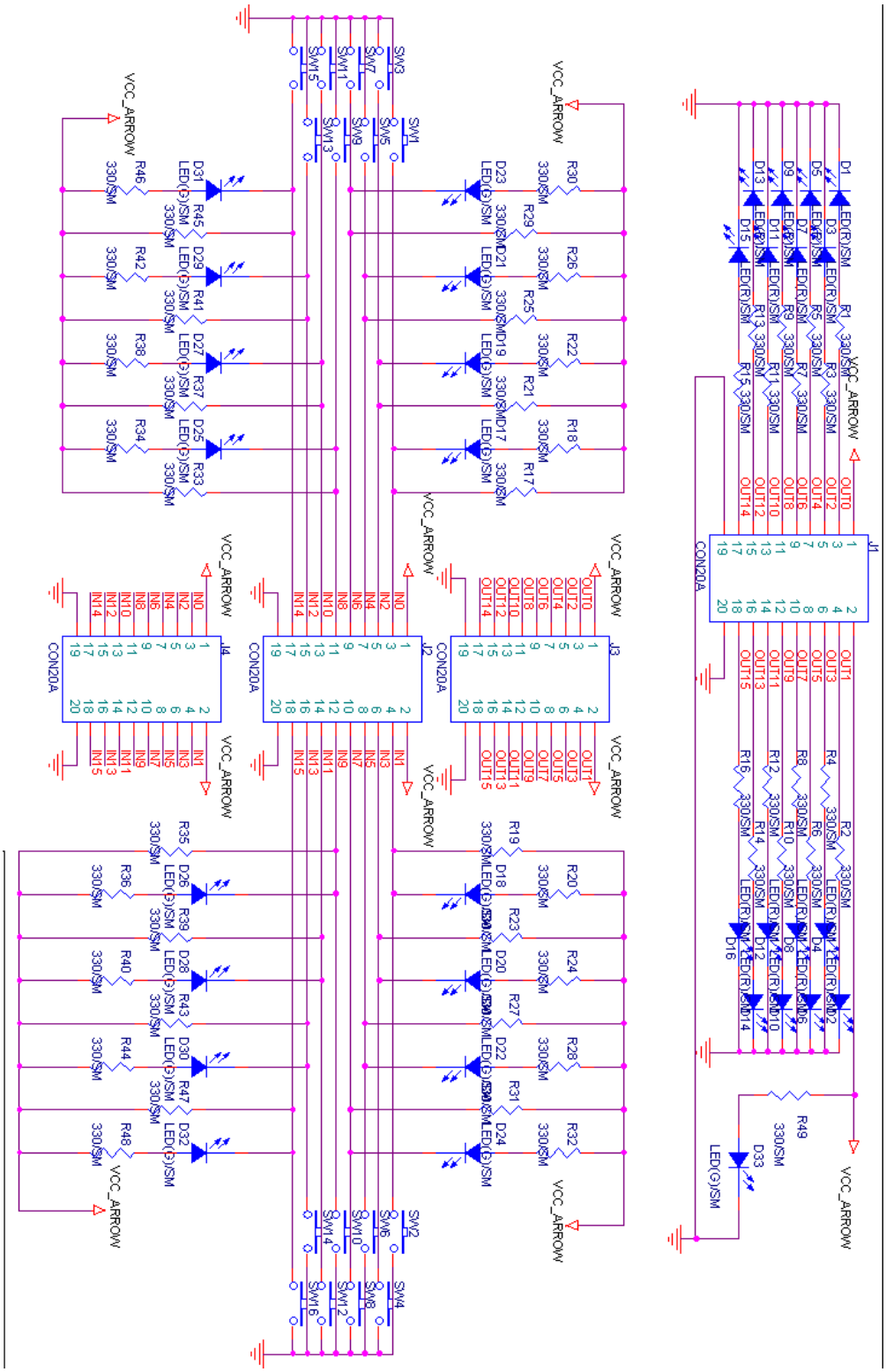


그림 6.6. HelloDevice 입출력 제어 보드의 회로도

6.2.2 사용자 장치의 하드웨어 구현

1) HelloDevice 와의 연결

실제로 HelloDevice 와 사용자 장치를 연결할 때는 데모보드를 사용하지 않고, 2.5 mm 피치의 20 핀 커넥터를 이용하여 직접 연결할 수 있다. 사용자가 구성한 하드웨어의 전원은 HelloDevice 로부터 20 핀 코넥터를 통하여 5V 전원의 형태로 공급받을 수 있으며, 사용자 자신의 전원을 따로 구성하여 사용할 수도 있다. HelloDevice 의 입출력 접점의 동작은 5V TTL 이며, 회로 구현 방식에 따라 HIGH 또는 LOW 로 입력을 인식하거나 출력 상태를 변경할 수 있다. 즉, 입력 또는 출력 접점의 상태는 장치의 특성에 따라, HIGH Active 또는 LOW Active 로 초기화하고 동작시킬 수 있다.

2) 입력부 설계

사용자 장치의 포토커플러를 통하여 HelloDevice 의 입력 인터페이스부에 5V TTL 레벨의 HIGH 또는 LOW 신호를 보낼 수 있다. 즉, 외부 장치의 입력 단자와 연결된 포토커플러 (PC345)가 ON 상태로 되었을 때, HelloDevice 의 입력 접점부에서는 HIGH 에서 LOW 로 떨어지는 신호를 입력 받을 수 있게 된다.

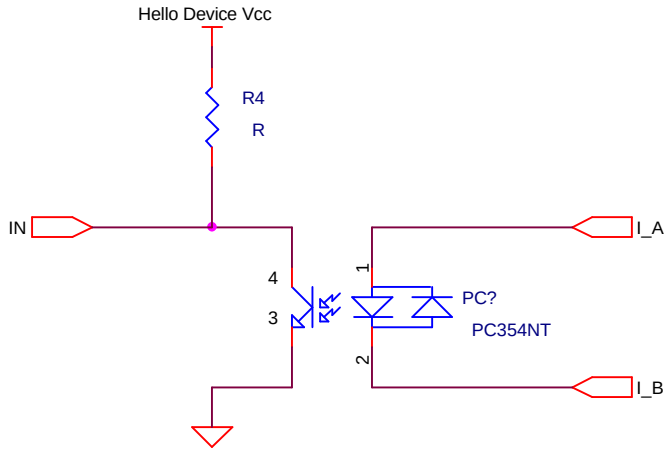


그림 6.7. HelloDevice 와의 입력 인터페이스 하드웨어 설계 예

3) 출력부 설계

HelloDevice 에서 출력된 5V TTL 레벨 신호는 사용자 장치의 전압 레벨에 맞게 신호를 받게 된다. HelloDevice 의 Vcc 와 출력 접점 사이에 포토커플러(pc817)를 연결하여, 출력 접점의 상태를 HIGH 에서 LOW 로 동작시켰을 때, 릴레이 동작에 의해 연결된 장치의 ON/OFF 제어가 가능하게 된다. 그림 6.8 은 출력 접점 사용을 위한 회로도의 예를 나타낸 것이다.

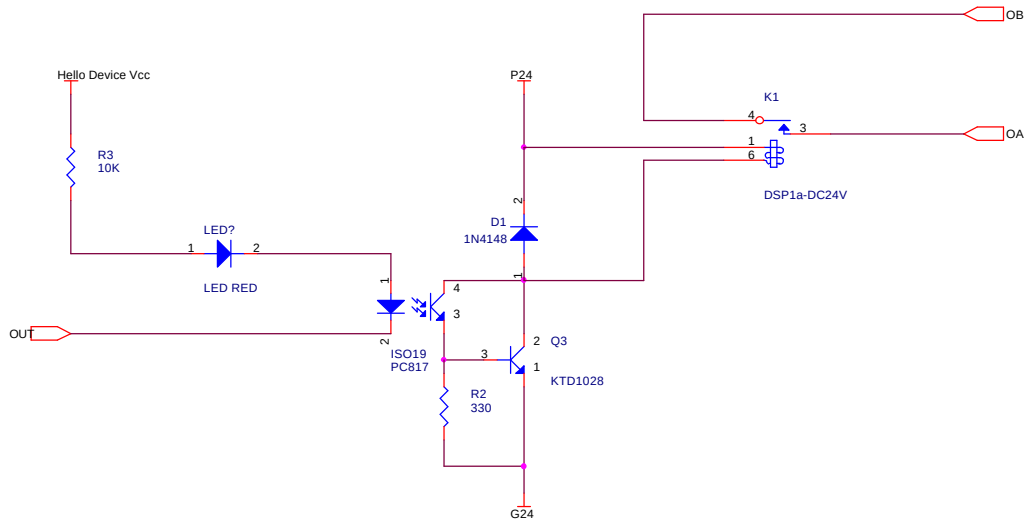


그림 6.8. HelloDevice 와의 출력 인터페이스 하드웨어 설계 예

6.3 소프트웨어 예제

6.3.1 소켓 프로그램을 이용한 통신

다음 C 프로그램 소스 코드는, HelloDevice 의 TCP 포트 6001 로 입출력 제어 신호를 전송하고 수신하는 인터넷 소켓 인터페이스를 버클리 소켓 C 함수로 구현한 예이다. 사용자는 이 소스 코드를 참고하여, 원격으로 입출력을 제어하는 응용 프로그램을 작성할 수 있다. 전체 소스 코드는 “HelloDevice 설치 폴더\Source\C\iodemo.c”이다.

```
//---------------------
// Process I/O Monitor
//---------------------
void IOMonitor()
{
    char commandBuf, *ResponseBuf ;
    int     commandLen, lenReceived ;
    int     i ;
    int err, clientLen ;

    // Initialize TCP socket
    TCPSocketInit() ;

    // Make TCP command
    commandBuf = IOGet ; // Command ID
    commandLen = 1 ;

    // Send command to HelloDevice
    err = sendto
            (
            sock,
            &commandBuf,
            commandLen,
            0,
            (struct sockaddr*)&clientAddr,
            sizeof(clientAddr)
            ) ;
```

```
    if (err == -1)
    {
        perror("\nsend error\n");
        exit (1);
    }

    // Allocate buffer for incoming packet = 16 bytes!!!
    ResponseBuf = calloc(0x10, sizeof(char)) ;

    // Receive incomming packet....
    lenReceived = recvfrom
                    (
                    sock,
                    ResponseBuf,
                    0x10,
                    0,
                    (struct sockaddr*)&clientAddr,
                    &clientLen
                    );

    if (lenReceived < 0)
    {
        perror("\nError receiving???\n") ;
        exit(0) ;
    }

    // Display incoming packet size
    printf("\n%d bytes received...\n", lenReceived) ;

    // Store I/O status for future use
    for (i=0; i<4; i++)
        IOStatus[i] = ResponseBuf[i] ;

    // Display I/O status
    printf("\n\n*********************************\n") ;
    printf("Input   :   %x:%x\tOutput   :   %x:%x",   IOStatus[0],   IOStatus[1],   IOStatus[2],
IOStatus[3]) ;
    printf("\n*********************************\n") ;

    // Free
    free(ResponseBuf) ;

    // Close TCP socket
    TCPSocketClose() ;
}



//---------------------
// Process Output set
//---------------------
void OutputSet()
{
    char    commandBuf[3] ;
    int     commandLen ;
    int     outbit=0, outdata=0x0001 ;
    int     err ;

    // Read output set value
    printf("Select the output point to be set (0-15) :") ;
    scanf("%d", &outbit) ;

    // Re-Initialize TCP socket
    TCPSocketInit() ;

    // Determine the output value considering current output status
    outdata = ((IOStatus[2]<<8) | IOStatus[3]) ;  // Read current output status
    outdata |= (int) (1 << outbit);       // Or operation with currentle selected Bit

    // Store current output status
    IOStatus[2] = (outdata & 0x0000ff00)>> 8 ;
    IOStatus[3] = (outdata & 0x000000ff) ;
```

```
    // Make TCP command

    // 1) Command ID
    commandBuf[0] = IOSet ;

    // 2) Output status set
    commandBuf[1] = (BYTE) ((outdata & 0x0000ff00) >> 8) ;
    commandBuf[2] = (BYTE) (outdata & 0x000000ff) ;

    commandLen = 3 ;

    // Send command to HelloDevice
    err = sendto
            (
            sock,
            &commandBuf,
            commandLen,
            0,
            (struct sockaddr*)&clientAddr,
            sizeof(clientAddr)
            ) ;
    if (err == -1 )
    {
        perror("\nsend error\n");
        exit (1);
    }

    // Close TCP socket
    TCPSocketClose() ;

}
```

6.3.2 자바 애플릿을 이용한 통신

자바 애플릿을 이용한 프로그램 소스 코드는 “HelloDevice 설치 폴더\Source\java” 폴더에 위치한 IOdemo.java, IOComm.java, Led.java, OutButton.java 이다. 사용자는 사용자의 웹 페이지를 제작하고, 이 자바 애플릿을 html 파일에 삽입하여 웹 기반의 원격 입출력 제어를 구현할 수 있다.
다음 자바 프로그램 소스 코드는, HelloDevice 의 TCP 포트 6001 로 입출력 제어 신호를 전송하고 수신하는 인터넷 소켓 인터페이스를 자바 애플릿으로 구현한 예이다.

```
    /* Output read */
    public int readValueTCP() {
        Socket socketTCP = null;
        int tmp = 0;
        int inputData = 0;
        byte rxData[] = new byte[16];
        byte data[] = {COMMAND_GET};
        try {
            socketTCP = new Socket(InetAddress.getByName(m_host), m_port);
            socketTCP.setTcpNoDelay(true);
            socketTCP.getOutputStream().write(data, 0, data.length);
            instream = new DataInputStream(socketTCP.getInputStream());
            tmp = instream.read(rxData, 0, rxData.length);
            if (tmp != -1) {
                inputData = (int) (rxData[2] << 8 | (rxData[3] & 0x00ff));
                inputData &= 0xffff;
            }
            socketTCP.close();
            instream.close();
```

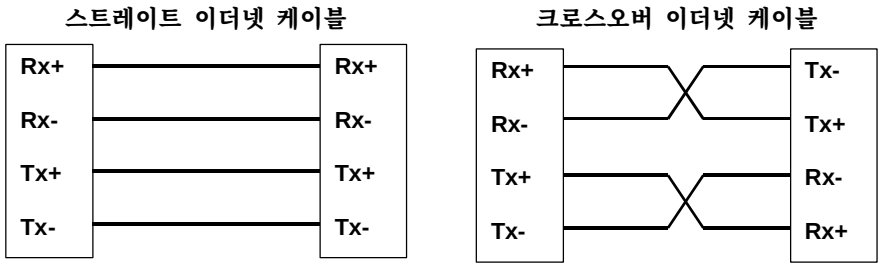
```
    } catch (Exception e) {
        System.out.println("Err : " + e);
    }
    return inputData;
}

/* Output write */
public void wirteValueTCP(int outdata) {
    Socket socketTCP = null;
    byte[] data = new byte[4];
    data[0] = COMMAND_SET;
    data[1] = (byte) ((outdata >> 8) & 0x000000ff);
    data[2] = (byte) (outdata & 0x000000ff);

    // Initialize socket
    try {
        socketTCP = new Socket(InetAddress.getByName(m_host), m_port);
        socketTCP.setTcpNoDelay(true);
        socketTCP.getOutputStream().write(data, 0, data.length);
        socketTCP.close();
    } catch (Exception e) {
        System.out.println("Err: " + e);
    }
}
```

Appendix A. 크로스오버(cross-over) 이더넷 케이블을 사용한 IP 주소 재설정

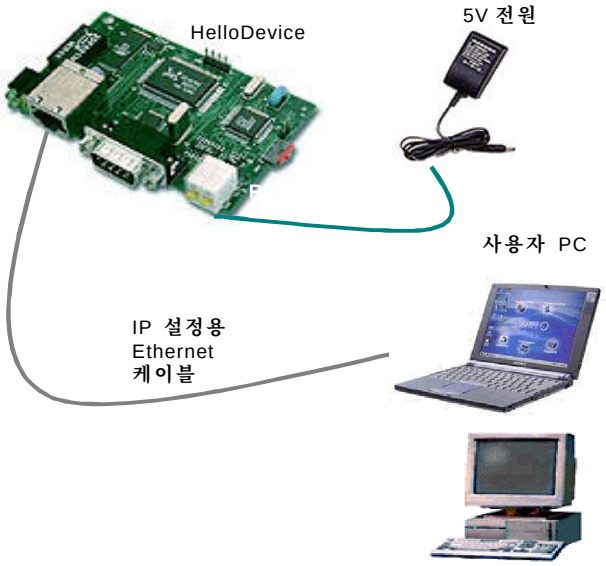
HelloDevice 의 IP 주소를 잘못 설정하여 현재 네트워크 환경과 맞지 않게 설정하였을 경우 (예: 사용자 네트워크 환경은 172.168.1.xxx 환경으로 설정되어 있는데, HelloDevice IP 주소를 192.167.1.23 으로 설정하였을 경우), 사용자 PC 에서는 네트워크를 통해 HelloDevice 와 통신할 방법이 없다. 이러한 경우에는 크로스오버 이더넷 케이블을 사용한다. 이 케이블은 Rx-Rx, Tx-Tx 를 연결한 보통의 스트레이트 (Straight) 이더넷 케이블이 아니고, Rx-Tx, Tx-Rx 를 연결한 것이다.



크로스오버 이더넷 케이블을 PC 와 HelloDevice 간에 직접 연결하면 네트워크 라우팅의 영향을 받지 않고, 사용자가 의도하는 임의의 IP 주소를 HelloDevice 에 할당할 수 있다. 사용 방법을 설명하도록 한다.

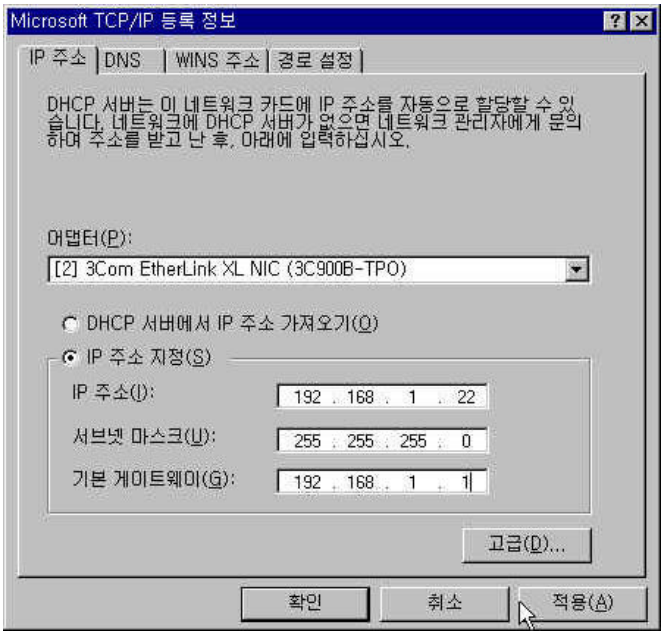
1) 이더넷 카드가 내장되어 있는 사용자 PC 에 연결된 기존의 이더넷 케이블을 제거한다.

2) HelloDevice 와 사용자 PC 간에 크로스오버 이더넷 케이블을 직접 연결한다.



3) 이와 같은 연결은, 1:1 연결이므로, 사용자 PC 의 네트워크 환경 설정을 HelloDevice 와 통신할 수 있도록 맞추어 주어야 한다. 네트워크 환경의 TCP/IP 프로토콜 설정에서의 사용자의 IP 주소 및 Gateway 주소를 임시로 HelloDevice 와 같은 서브넷 마스크 상에서 동작하도록 조작해 준다.

예) HelloDevice 의 IP 주소가 192.168.1.23 으로 설정되어 있고, 현재 사용자 PC 의 IP 주소가 172.168.1.22, 서브넷 마스크가 255.255.255.0 일 경우, 사용자 PC 의 TCP/IP 프로토콜 설정을 다음 그림과 같이 HelloDevice 와 통신할 수 있는 형태로 IP 주소가 1 만큼 차이가 나도록 192.168.1.22 로 맞추어 주고 [확인] 버튼을 누르도록 한다. 이렇게 하면, 현재 상태에서 IP 주소가 192.168.1.23 인 HelloDevice 와 통신할 수 있게 된다. 이 설정은 임시적인 설정이므로, 사용자는 원래의 네트워크 환경에서의 IP 주소 등의 네트워크 환경 설정을 기억해 놓도록 한다.



4) 네트워크 설정을 확인하기 위해, ping 을 사용하여 다음과 같이 시험해 본다. 다음과 같이, 응답이 오면 네트워크 설정이 적절하게 된 것이다.

```
>> ping 192.168.1.23
>> Pinging 192.168.1.23 with 32 bytes of data:
      Reply from 192.168.1.23: bytes=32 time=10ms TTL=251
      Reply from 192.168.1.23: bytes=32 time<10ms TTL=251
      Reply from 192.168.1.23: bytes=32 time=10ms TTL=251
```

5) HelloDevice 유틸리티 소프트웨어를 실행한다.

6) 4.3.2 **절**, [IP **주소 변경]에서 설명된 대로**, IP **주소를 사용자의 의도대로 재설정한다**.