

산업용 6관절 로봇의 원격제어를 위한 실시간 병렬데이터통신 인터페이스

Development of Realtime Parallel Data Communication Interface for Remote Control of 6-DOF Industrial Robot

최 명 환* 이 우 원**
Choi, Myoung-Hwan Lee, Woo-Won

Abstract

This paper presents the development of the I/O Interface for the real time parallel data communication between controller of a six-axis industrial robot(CRS-A460) and an external computer. The proposed I/O Interface consists of the hardware I/O interface and the software that is downloaded to the robot controller and executed by the controller operating system. The constitution of the digital I/O Port for CRS-A460 robot controller and the digital I/O board for IBM-PC are presented as well as the Process Control Program of the robot controller. The developed protocol for the parallel data communication is described. The data communication is tested, and the performance is analysed. In particular, it is shown that the real-time constraint of the robot controller process is satisfied.

키워드 : 산업용 로봇, 병렬통신, 인터페이스

Keywords : 6-DOF Industrial Robot, Parallel Data Communication, Interface

1. 서 론

현대에는 산업용 로봇을 이용한 다양한 작업이 이루어지고 있다. 공장환경에 따라서 다양한 산업용 로봇이 있는데, 주로 사용되는 산업용 로봇은 조립 생산라인과, 반도체생산라인, 자동차 용접라인 등 다양한 생산 및 조립 라인에서 사용되고 있다. 특히 많이 사용되는 로봇은 수평 다관절 로봇과 수직 다관절 로봇을 예로 들 수 있다.

이러한 산업용 로봇의 동작을 제어하고, 원하는 작업을 수행할 수 있도록 하는 것이 로봇교시방법

이다. 로봇에게 필요한 작업을 지시하기 위해서는 로봇의 제어기에 부착되어 있는 교시상자를 이용하거나, 혹은 직접 로봇 Tool에 부착되어 있는 힘/모멘트 센서, 혹은 교시상자에 붙어있는 6D Mouse를 이용한다.

일반적으로 산업용 로봇의 제어기는 제어기 내에 로봇의 운동에 필요한 하드웨어 및 소프트웨어를 모두 내장하고 있으며, 독립적으로 동작한다. 만약, 로봇 제어기의 외부에서 로봇제어기가 제공하지 않는 기능을 수행하도록 로봇을 제어하고자 한다면, 대부분의 로봇에서는 로봇제어장치의 심각한 교체 혹은 수정 없이는 그 작업은 불가능하다. 일부 로봇제어기의 경우는 로봇제어기가 제공하지 않는 기능을 수행할 수 있도록, 외부에서 로봇을 원격제어 할 수 있도록 하는 기능을 포함하고 있

* 강원대학교 전기전자정보통신공학부 교수, 공학박사
** 강원대학교 대학원 제어계측공학과 석사과정

다. 본 논문에서 사용한 CRS A460 로봇의 경우, 외부에서 입력한 운동지시명령에 따라 로봇을 제어할 수 있게 하는 명령어 REMOTE 가 있으며 이를 이용하여 원격 컴퓨터가 로봇을 제어할 수 있다. 문제는 운동지시명령을 실시간으로 REMOTE 명령어로 전달하는 방법이 없다는 것인데, 로봇제어기의 디지털 I/O 는 단순한 제어신호의 I/O 에 사용되는 것으로, REMOTE 명령어의 실시간 전송에 필요한 매 32msec 샘플타임 마다 6 개 관절의 위치명령 값, 즉 24 byte 의 전송이 불가능하다. 따라서 본 논문의 내용은 이러한 데이터를 외부 PC에서 로봇제어기로 실시간으로 전송하기 위한 하드웨어 및 소프트웨어 인터페이스를 설명하고 있으며, 하드웨어 인터페이스는 PC-ROBOT 제어기간의 5byte 양방향 데이터 통신 회로가 주 내용이며, 소프트웨어 인터페이스는 PC-ROBOT 제어기간의 통신프로토콜의 자체 개발과, 로봇 운영체제 상에서 동작하면서 I/O 데이터를 읽어들이어 REMOTE 명령어로 전달하는 I/O 드라이버 성격의 프로그램 모듈의 개발에 대한 부분이 주 내용이 되고 있다.

본 논문에 사용된 CRS-A460로봇은 제어기 내부에 5개의 디지털 입/출력 Port를 내장하고 있다. 따라서, 이 Port를 이용하여 외부로부터 데이터를 받고, 보낼 수가 있다. 또한, 제어기에 사용자가 개발한 S/W를 Download하여 실행시킬 수 있는 방법이 제공되므로, 이 Port를 이용하여 받아들인 데이터를 제어기에서 직접 제어할 수도 있고, 사용자의 원하는 작업에 따라서 로봇 운영체제를 거치지 않고 직접 I/O 포트의 접근이 가능하다. 이러한 특징을 이용하여 본 논문에서는 기존에 저자에 의해서 제작된 힘/모멘트 센서를 이용하여 산업용 로봇인 CRS-A460로봇을 원격제어 함으로서 교시작업을 수행하고자 한다. 이러한 교시작업을 수행하기 위해서는 제작된 힘/모멘트 센서로부터 발생하는 신호를 로봇의 각 Joint에 관한 데이터로 변환시켜서 로봇으로 전달하여 동작시킬 수 있는 Interface가 필요하게 된다. 이러한 Interface는 힘/모멘트 센서에서 발생하는 X, Y, Z, Yaw, Pitch, Roll 신호를 짧은 시간 안에 모두 로봇 제어기에 전달시켜야만 한다. 그리고, 전달되는 신호는 실시간으로 전달되게 됨으로써 전송 중에 오류가 발생할 수도 있는데, 이러한 오류가 없이 데이터가 올바르게 전송되어야만 한다. 따라서, 이러한 데이터의 전송을 위하여 본 논문에서는 산업용 6관절 로봇인 CRS-A460로봇과의 병렬 데이터 통신을 위한 인터페이스 개발을 소개한다. 소개되는 인터페이스는 H/W 부분과 하드웨어를 인터페이스를 동작시켜 데이터의 전송을 원활히 할 S/W Interface의 두 부분으로 나누어서 소개하고, 완성된 Interface에 대한 전송 테스트 및 분석을 한다.

2. 병렬 I/O 인터페이스 하드웨어

본 논문에서 제작한 병렬 통신용 하드웨어 Interface 구성도는 그림 1에서 보여준다. 그림에서 보듯이 IBM-PC 내부에 있는 6-Channel Digital I/O와 Robot Controller 내부의 5개의 Digital I/O Port를, 제작한 H/W Interface 보드가 연결되어 IBM-PC와 Robot Controller사이의 병렬 데이터를 전달해준다. 그림 2는 PC의 Digital I/O Board와 Robot의 Digital I/O Port사이에서 병렬 데이터를 전달해주는 하드웨어 Interface회로에서 하나의 Photo-coupler를 이용하여 데이터를 전달하는 회로이다. 회로의 특징은 Photo-coupler를 사용하여 IBM-PC와 Robot Controller 사이의 전원을 분리시키고, 순수하게 데이터 신호만이 전달될 수 있도록 한 것이다.

회로 왼쪽에는 PC Digital I/O Board의 출력 Port가 위치하며, 출력 신호는 Photo-Coupler를 거쳐 회로도 오른쪽의 Robot 제어기 Digital I/O Port로 전달된다. 전달되는 신호는 PC에서 로봇으로 Data 24bit, 제어신호 8bit가 출력되며, 로봇 제어기에서 PC로 Data 24bit와 8bit의 제어신호가 출력된다. PC에서 로봇으로, 혹은 로봇에서 PC로 데이터를 전송하는 회로는 동일한 구조를 갖는다.

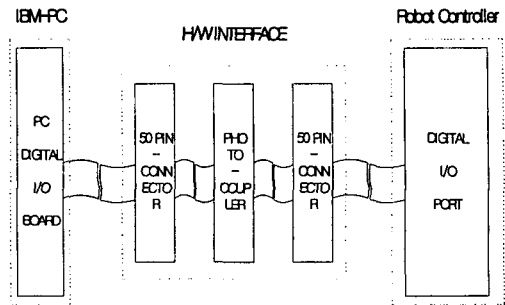


그림 1. 병렬 I/O 하드웨어 Interface 구성도

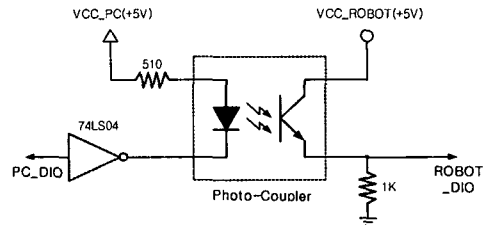


그림 2. PC와 Robot의 Data전송을 위한 Interface회로의 Photo-coupler회로

그림 3은 제작된 하드웨어 Interface로써 IBM-PC와 Robot 제어가 50pin 병렬 Cable들로 연결되어있다. 그림 3에서 하드웨어 인터페이스는 세 부분으로 나눌 수 있다. 위쪽부분의 connector 단자가 PC의 I/O Board와 연결되는 부분이고, 아래의 두 개의 병렬 cable이 연결된 connector들은 Robot의 I/O Port와 연결되는 부분이며, 가운데 부분은 위쪽 부분과 아래쪽 부분의 전원을 분리하기 위한 소자들(Photo-coupler 등)이 위치한 부분이다.

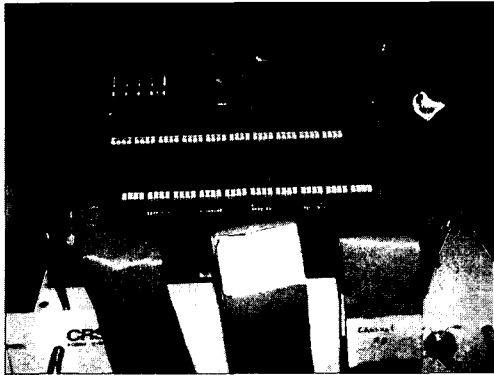


그림 3. PC-Robot Interface 회로

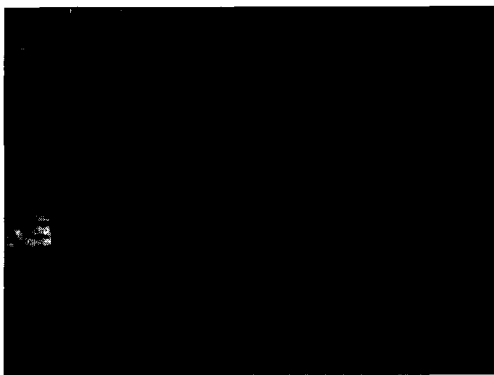


그림 4. Robot Controller의 Digital I/O Ports

본 논문에서 제시한 산업용 로봇인 CRS A460 로봇의 controller에는 총 5개의 Digital I/O Port가 있으며, 그림 4의 아래 부분부터 Port1, 2, 3, 4, 5의 순서로 되어있다. 로봇 제어기 Digital I/O Port중 Port 하나 당 입력 8bit, 출력 8bit를 가지고 있으며, 총 5개의 Port가 존재한다. PC의 Digital I/O Board는 총 6개의 Channel로 구성되어 있는데, 각 채널은 2개의 출력 Port와 1개의 입/출력 Port를 가지고 있다. 기본적으로 통신은 PC의 Digital I/O board의 출력은 로봇 제어기의 입력 Port로 연결되고, 로봇 제어기에서는 입력 Port를

통해 읽어들이는 값을 다시 로봇 제어기의 출력 Port로 내보낸다. 이 출력 값은 PC Digital I/O Board의 입력 Port로 return되며, PC의 Digital I/O Board의 입력 Port를 통해 읽어들이는 값을 로봇 제어기로 출력한 data와 비교하여 PC와 로봇사이의 데이터 통신에 이상이 없는지를 확인한다.

3. 병렬 I/O 인터페이스 소프트웨어

3.1 PC(Process Control Programs) 프로그램

본 논문의 Interface 동작에 사용되는 로봇 Controller에는 Process Control Programs (PCP)라고 하는 기능이 있는데, 이것은 로봇 Controller를 External process를 사용하여 직접 제어할 수 있도록 하는 기능이다. 따라서, 이러한 기능을 이용하면 CRS-A460의 로봇 제어 언어(RAPL-II)를 사용하지 않고서도 로봇을 동작시킬 수 있게 된다. PCP는 외부 PC에서 Robot이 수행하기를 원하는 기능을 C 언어로 프로그램하여 8086 machine code로 Cross-compile한 뒤 로봇 controller에 download 시켜서 수행시킨다. PCP는 로봇에서 이용 가능한 함수들을 사용하기 위해서 로봇 controller의 자원들을 자유롭게 이용할 수 있다. 이러한 PCP 프로그램의 사용은 일반적으로 사용하는 방법이 아니며, 제어기의 메모리에 상주하는 운영체제의 동작을 방해할 수 있으므로, 사용상의 주의가 필요하다. PCP 프로그램을 로봇 controller에 download 하기 위해서는 가장 일반적인 RS232C Serial port를 사용한다. 그리고, Robot의 운동을 발생하기 위해서는 Robot 언어에서 제공하는 REMOTE명령어를 사용한다.

3.2 로봇 운동에 필요한 PCP용 함수

이러한 PCP 프로그램을 사용하여 로봇 제어기의 Digital I/O Port로부터 데이터를 읽고 쓰기 위해서는 직접 입/출력 Port에 데이터를 읽고 써주는 C언어에서 기본적으로 제공하는 inport(), inportb(), outport(), outportb() 함수를 사용한다. 그리고, 로봇 제어기에서는 32ms마다 한번씩 로봇의 Digital I/O Port로 데이터를 써주는데, 이렇게 데이터를 써주는 버퍼에 원하는 값을 넣기 위한 루틴이 필요하게 된다. 아래의 박스에서 보여주는 부분은 이러한 입/출력 Port를 제어하기 위한 프로그램의 사용 예이다. 박스의 부분은 로봇 제어기의 Digital I/O Port로 제어신호 bit 및 4bytes Data를 써주기 위한 부분으로써, 로봇 제어기에서는 32msec 마다 한번씩 제어기 내부에 있는 I/O 버퍼에서 32bit long형의 데이터를 Digital I/O Port로 써주게 된다. 직접 I/O Port로 데이터를 써주는 outport()함수만을 사용하여 I/O Port를 제어

할 경우에는 32msec 마다 I/O Port가 Update되는 이유로 정확한 값이 I/O Port를 통해서 전달되지 않게 된다. 따라서, 박스에서 보여주는 것처럼 먼저 32msec마다 I/O Port를 Update 하는 버퍼에 보내고자 하는 Data를 넣고, 직접 I/O Port를 통해서 Data를 전달하는 output()함수를 사용함으로써 정확한 Data를 전송할 수가 있다. 이와 같은 방법으로 제어신호와 4bytes Data를 PC로부터 원격으로 받을 수 있으며, 다시 역으로 PC로 원격으로 데이터를 보낼 수가 있다.

```

/* 32ms마다 로봇 제어기에서 I/O Port에 Data를 써 넣기 위한 부분 */
{
    // digout으로 I/O Port의 현재 상태를 읽어들인다.
    disable();
    ptr=(char*)((unsigned long*) MK_FP(0,0x00FC));
    digout = (unsigned long) ((unsigned long*)ptr)[6];
    enable();

    // 읽어들이 I/O Port의 상태에서 원하는 bit만을 0 또는 1로 설정(제어신호)하여 digout에 저장.
    digout = (digout | 0x00810000) & 0xff81ffff;

    // 제어 bit만 바뀐 digout의 값을 I/O Port에 써준다.
    disable();
    ptr=(char*)((unsigned long*) MK_FP(0,0x00FC));
    ((unsigned long *)ptr)[6] = digout;
    enable();
}
/* I/O Port에 직접 Data를 써 넣는다 */
outputb(0x804,0x81);
    
```

3.3 REMOTE 명령어

REMOTE 명령어는 Robot 제어기의 외부에서 지시한 운동 명령을 받아 Robot을 구동하도록 하는 명령어이다. Robot의 운동 명령은 상대위치와 절대위치의 형태일수 있고, 또한 Cartesian 좌표계와 Joint 좌표계에서 운동을 지시할 수 있다. 명령어의 사용 형태는 다음과 같다.

```

REMOTE <Cartesian/Joint>,<Absolute/Incremental>
    
```

첫 번째 Argument는 REMOTE 명령어가 절대/상대위치 인지를 나타내고, 두 번째 Argument는 REMOTE 명령어 기준 좌표계가 cartesian 인지 joint인지를 나타낸다.

Robot의 운동명령에 필요한 데이터는 힘/모멘트 센서의 검출 정보를 이용하여 PC에서 발생되며, PC의 Digital I/O Board를 통해 Robot 제어기의 I/O Port로 전달된다(그림 1). PC 프로그램은 Robot I/O Port에 입력된 운동명령 데이터를 읽어 들인 후 REMOTE 명령어가 사용할 수 있도록 제어기 메모리 상의 지정된 위치에 저장한다. 그리고, 이렇게 저장된 위치 데이터는 Interface 구동 프로그램에 의해서 Robot Controller의 REMOTE

명령어와 Motion_remote() 함수로 입력받은 데이터만큼 로봇을 움직이는데 사용된다.

Digital I/O Port를 통해서 전달된 로봇의 운동 명령 Data들은 PCP 프로그램에 의해서 제어기 메모리의 지정된 위치에 저장되고, 다음과 같은 순서로 이 로봇이 구동된다. 박스의 (1)과 같이 REMOTE 명령어와 함께 CART(Cartesian Mode)와 INC(Incremental Mode)를 Command 상태에서 입력하면 command 표시가 'R>'로 바뀌는데 이는 REMOTE 상태를 나타낸다. 그런 뒤 EXECUTE 명령어를 이용하여 PCP 프로그램이 저장된 Address를 (2)와 같이 지정하면 PCP 프로그램이 수행되면서 Interface를 통해서 데이터를 주고받으면서 로봇을 동작시키게 된다. (2)의 24880은 PCP 프로그램이 저장된 곳의 Address를 나타낸다.

```

J>REMOTE CART, INC  ▢ ..... (1)
R>
R>EXECUTE PHYSICAL ADDRESS : 24880  ▢ ..... (2)
    
```

아래의 박스는 PCP 프로그램의 일부로써, 제어기 메모리에 저장된 Data들을 이용하여 로봇을 구동시키는 함수이다. while문의 Data_ready_flag는 로봇 제어기가 로봇을 구동시키기 위한 Data 버퍼에 Data가 있는지 없는지를 알려준다. Data가 입력되거나 제어기가 사용될 준비가 되면 이 Data_ready_flag는 1로 되고, 이 때 로봇 제어기는 버퍼에서 데이터를 가져가서 로봇을 구동시킨다. 그리고, 제어기가 데이터를 가져가면 Data_ready_flag는 0(clear)으로 된다. 구동 버퍼가 clear되면 I/O Port로부터 입력받은 위치 데이터를 구동 버퍼에 넣고, (3)과 같이 다시 Data_ready_flag를 1로 만들어 줌으로써 제어기가 구동 버퍼의 데이터를 가져가서 로봇을 구동시키도록 한다.

```

// PCP용 프로그램의 함수로써, 입력된 Data를 이용하여 로봇을 구동시킨다
void Motion_remote(void)
{
    while(*Data_ready_flag!=0){ // 버퍼에 데이터가 있으면 clear될 때까지 기다린다.

        // 입력된 값을 구동용 버퍼에 저장한다.
        *cart_x = S_buff[0];
        *cart_y = S_buff[1];
        *cart_z = S_buff[2];
        *cart_yaw = S_buff[3];
        *cart_pitch = S_buff[4];
        *cart_roll = S_buff[5];

        // 구동버퍼에 데이터가 있다는 것을 제어기에 알려준다.
        *Data_ready_flag=0x01; ..... (3)
    }
}
    
```

이러한 방법을 통해서 Interface를 거쳐 로봇 제어기의 Digital I/O Port로 전달되는 로봇의 위치

데이터를 REMOTE 명령어와 PCP 함수를 통해서 사용함으로써 원격으로 로봇을 구동시킬 수가 있다.

PCP함수에서는 이렇게 REMOTE명령어와 Motion_remote()함수에 의해서 로봇을 구동시킨 후에 다시 구동이 끝난 로봇의 위치 데이터를 읽어 들여 데이터를 받을 때와는 반대로 다시 PC로 데이터를 보냄으로써 PC에서는 보낸 위치 data와 받은 위치 data를 비교 분석하여 전송이 제대로 되는지를 확인한다.

3.4 PC와 산업용 6관절 로봇 Data 전송 프로토콜

로봇 제어기에서는 Digital I/O Port를 이용한 병렬 통신 Protocol을 내부적으로 제공하지 않으므로, Digital I/O Port를 사용하기 위해서는 앞서 언급한 PCP프로그램을 통해서 병렬 데이터 전송을 위한 Protocol을 작성해야만 한다. 로봇 Controller에서는 내부적으로 32ms마다 한번씩 Digital I/O Port를 사용할 수 있기 때문에 이 시간 안에 4Byte의 병렬 데이터를 주고받아야만 한다. 이를 위해서는 병렬 데이터를 32ms안에 주고받도록 하는 Protocol이 필요하다. 따라서, 본 논문에서는 이를 위한 통신 Protocol을 Interface와 함께 제작하였다.

그림 5는 PC에서 하드웨어 병렬 I/O Interface를 통해서 로봇으로 병렬 데이터를 전송시키기 위한 설계된 데이터 전송 프로토콜의 Flow Chart를 보여준다.

그림 5에서 왼편의 Flow Chart가 Robot 제어기 쪽 프로그램(PCP)에서 데이터를 받는 부분이고, 오른편의 Flow Chart는 PC쪽 프로그램에서 데이터를 Robot 제어기로 보내는 부분이다. 그림 5에서의 PC->Robot Data 전송에 사용되는 제어신호(Send_DATA1)로는 1 Byte(8bit) 신호 중 LSB 1bit만을 사용한다. 이와 같은 전송 방법이 Robot->PC Data 전송 알고리즘에도 사용되었는데, Robot->PC Data 전송에 사용되는 제어신호(Send_DATA2)로는 1 Byte(8bit) 신호 중 MSB 1bit만을 사용한다. 4 Bytes(32bit)의 데이터가 전송되는 시점은 이 제어신호가 상승되는 edge 시점에서 서로 이루어진다. 표 1은 이러한 전송프로토콜에 따른 데이터의 전송 관계를 나타낸다. 그림 6은 PC<->Robot Data 전송 packet을 나타낸다. 그림 6과 같은 구조로 데이터를 전송하고 또한 검증한다. 이와 같은 전송 프로토콜을 사용하여 데이터를 전송시켰을 시에 PC<->Robot Data 전송 시간은 약 20ms 가량 나왔다. 따라서, 로봇에서 32ms 이하의 시간을 요구하므로, 요구조건을 충족함으로써 데이터 전송 프로토콜은 시간적으로 이상이 없음을 나타낸다.

표 1. PC<->Robot Data 전송 프로토콜에 따른 전송 데이터와 신호

전 송 프로토콜	PC->ROBOT Data 전송	ROBOT->PC Data 전송
Data (Bytes)	4 Bytes	4 Bytes
제어신호 (Byte)	1 Byte	1 Byte
사용-bit	LSB 1bit	MSB 1bit

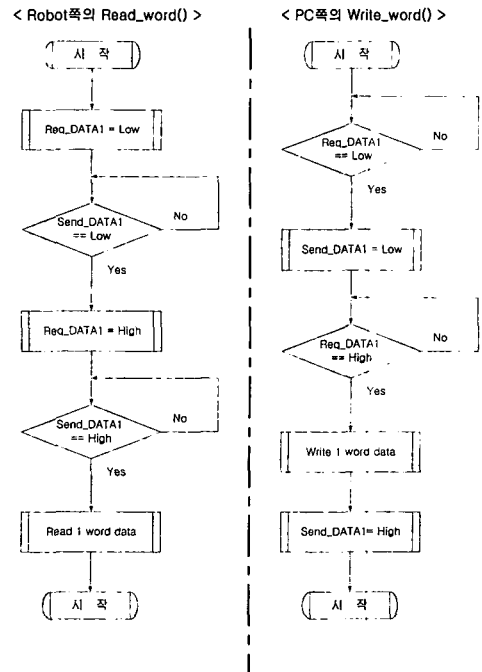


그림 5. PC->Robot Data 전송 알고리즘



그림 6. PC<->Robot Data Packet (8 word = 32 Bytes)

4. PC<->Robot Data 전송 프로그램 테스트

본 논문에서 보여준 병렬 데이터 통신을 위한 인터페이스의 테스트를 위해서 그림 1과 같이 인

터페이스 회로를 연결한 후, 로봇 제어기에 PCP 프로그램을 수행시키고, PC에서는 데이터 전송 테스트 프로그램을 수행시킨다. IBM-PC쪽에서는 임의의 Float값(힘/모멘트 센서 값)을 정하고, Robot Controller쪽에서는 임의의 로봇의 위치 데이터값(역시 Float값)을 정한다. 이렇게 정해진 데이터를 PC->Robot 상호간에 작성된 병렬 통신 알고리즘이 적용된 프로그램에 의해서 테스트를 실시한다. 먼저 IBM-PC에서 테스트 데이터를 병렬 통신 I/O Data packet에 맞추어서 보내고, 바로 수신 모드로 바꾼다. 그러면, 이 보내지는 데이터를 Robot PCP프로그램에서는 병렬 통신 I/O Data packet에 따라서 전송여러 여부를 확인한다[그림 6]. 확인 작업이 끝나면 이어서 Robot PCP프로그램에서는 내부의 Digital I/O Port를 통해서 IBM-PC로 로봇의 위치 데이터(X, Y, Z, θ_z , θ_y , θ_x)를 보낸다. 그러면, 수신모드로 되어 있던 IBM-PC에서는 이 데이터를 받아서 올바른지를 확인한 뒤 그 결과를 모니터에 Display 한다. 이렇게 1개의 packet을 주고받는 작업이 완료되면 하나의 전송 Cycle이 된다.

이러한 데이터의 전송 상태를 오실로스코프를 이용하여 파형을 잡아내고, 이를 분석해 보았다. 그림 7은 1 Cycle의 PC->ROBOT Data 전송 프로그램 테스트 중에 PC->Robot으로 보내지는 'Send_DATA1' Signal 과 Robot->PC 로 보내지는 'Send_DATA2' Signal 이다. 그림 7에서 윗쪽에 있는 파형이 PC->Robot의 제어신호이고, 아래쪽에 보이는 파형이 Robot->PC 제어신호 이다. 이 때의 1 Cycle이 모두 걸리는 시간은 9.77193ms 가 나왔다.

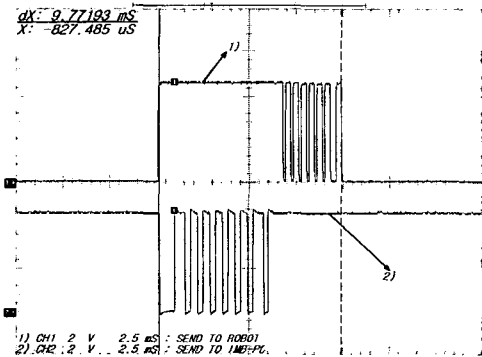


그림 7. 1-Cycle의 PC->ROBOT Data 전송 시 측정된 Send_DATA1과 Send_DATA2 제어신호

그림 8은 연속적으로 데이터가 전송되는 PC->Robot Data 전송 제어신호(Send_DATA1, Send_DATA2) 파형을 나타내었다. 그림 8에서 보면 PC->Robot으로 하나의 Data packet(32Bytes)

을 보내는데 걸리는 시간은 3.14286ms 가 나왔고, Robot->PC로 하나의 Data packet을 보내는데 걸리는 시간은 5.8797ms 가 나왔다. 이렇게 두 제어신호의 시간이 차이가 나는 원인은 IBM-PC에서 처리하는 내부 속도가 로봇 제어기 내부에서 처리하는 속도보다 빠르기 때문에 이와 같은 현상이 나타난다.

따라서, 그림 8에서처럼 두 제어신호사이에 걸리는 시간을 파형에서 찾아보면 9.03759ms 가 나왔다. 이 시간은 IBM-PC와 로봇 제어기 내부에서 사용되는 시간에 의해서 똑같은 주기를 갖지는 않으나, 비슷한 값으로 나오는 것을 볼 수 있다.

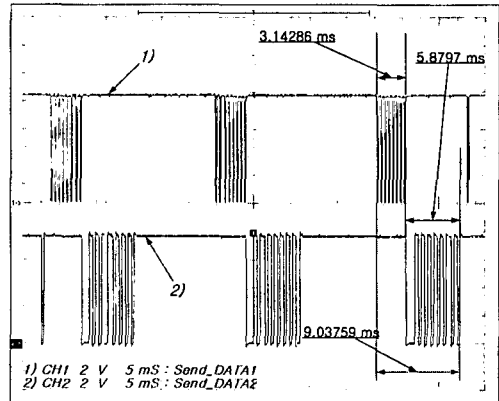


그림 8. 연속된 데이터 전송 시 PC->Robot Data 제어신호(Send_DATA1, Send_DATA2) 파형

그림 9는 앞서 보여준 파형들처럼 연속적으로 데이터가 전송되는 PC->Robot Data 전송 제어신호 파형을 나타내었다. 그림 9에서는 Robot->PC Data 전송 제어신호가 연속적으로 전송 될 때 하나의 packet이 전송되고 다음 packet이 전송되기 시작하는 시점까지의 시간, 즉 얼마의 주기를 가지는지 알 수 있다. 그림에서 보면 주기는 약 17.1579ms가 나왔다. 본 논문에서 사용된 산업용 6관절 로봇(CRS-A460)은 내부적으로 32ms의 주기를 가지고 제어기 내부 작업을 수행한다. 따라서, 이 주기 안에 1-Cycle의 병렬 I/O packet이 전송되어야만 하는 것이다. 그런데, 실험에서 측정 한 시간에 따르면, 전송 주기가 17.1579ms를 가지므로, REMOTE 명령을 사용한 로봇의 동작 명령이 소요되는 시간을 더하다라도, 32ms 시간 안에 모든 동작을 완료 할 수가 있다. 그러므로, 실시간으로 하드웨어 Interface를 통한 병렬 I/O 데이터 전송이 가능하고, 전송된 힘/모멘트 센서 데이터를 이용하여 산업용 6관절 로봇을 구동시킬 수가 있다.

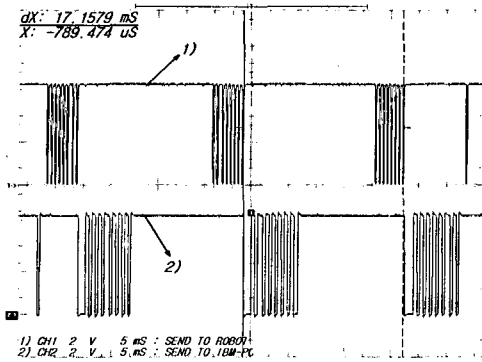


그림 9. 병렬 데이터의 전송 주기시간 측정결과

5. PC<->Robot Data 전송 프로그램 테스트 결과

지금까지 앞서 언급한 프로그램 테스트에서 PC에서 로봇 제어기로 보내게 되는 로봇 위치 Data X, Y, Z, θ_z , θ_y , θ_x 의 값 중에서 X 값을 매 32msec 샘플마다 +0.5mm로 반복해서 전송 테스트 실시 해 본 결과 로봇의 Tool이 Tool 좌표 시스템상의 X축 방향으로 +0.016m/sec의 속도로 움직였다. 또한 θ_x 에 매 32msec 샘플마다 +0.05 rad의 값으로 반복해서 전송 테스트를 실시해 본 결과 1.57rad/sec의 속도로 회전하였다. 이와 같은 실험 결과로 볼 때 본 논문에서의 병렬 데이터 통신 인터페이스를 통해서 실시간으로 로봇을 제어할 수가 있다.

6. 결론

본 논문에서는 힘/모멘트 센서신호를 이용한 산업용 로봇인 CRS-A460 로봇을 교시하기 위한 병렬 데이터 통신 Interface의 하드웨어 및 소프트웨어적인 특징과 병렬 데이터의 전송 알고리즘, 완성된 Interface에 대한 병렬 I/O Data 전송 실험에 대해서 살펴보았다. 힘/모멘트 센서 데이터를 CRS-A460 로봇의 교시작업에 사용될 수 있도록 하기 위하여 병렬의 센서 데이터를 실시간으로 로봇에게 전달해야만 하는 목적에 의해서 병렬 데이터 전송을 위한 하드웨어 Interface 보드에 대한

설계와 CRS-A460 로봇 및 IBM-PC에서 사용되는 Digital I/O 시스템에 대한 구성을 살펴보았고, 제작된 하드웨어 Interface를 통해서 그림 6에서 보여주는 I/O packet을 전송하고 전송 받기 위한 구동 프로그램에 대해서도 살펴보았다.

이렇게 제작된 병렬 I/O Data 전송 Interface 시스템의 성능을 분석하여 실시간으로 Robot에 Data를 전송할 수 있는지를 살펴보기 위한 방법으로 임의의 병렬 데이터들을 생성하여 전송시켜보고, 전송되는 데이터의 파형을 검출하여 살펴보았다. 하나 및 연속된 병렬 Data I/O packet에 대한 PC->Robot Data 전송 시간은 3.14286ms가 나왔고, Robot->PC Data 전송 시간은 5.8797ms가 나왔다. 그리고, 연속되는 전송 Data의 파형에서 로봇에서 데이터의 전송에 사용되는 시간이 17.1579ms로 CRS-A460 로봇에서 요구되는 32ms 이하의 시간이 측정결과로 나왔으므로 이것은 실시간으로 병렬 데이터를 제작된 Interface를 통해서 전송할 수 있다

참고 문헌

- [1] MOTOMAN-ET System, 安川電機, Japan, 1995
- [2] Gerd Hirzinger, "Advances in Robotics - An European Perspective", Plenary Lecture, IEEE International Conference on Robotics and Automation, May, 1998, Leuven, Belgium
- [3] 최명환, "COSMO-로봇 교시를 위한 저가형 힘/토크 센서", 97 한국자동제어학술회의, pp. 1621-1624, 서울, 1997
- [4] 최명환, "초저가형 힘/모멘트 센서 COSMO를 이용한 직관적 교시방법", 98 한국자동제어학술회의, 부산, 1998
- [5] JOHN J. CRAIG, "Introduction to ROBOTICS : Mechanics and Control second edition", Addison-Wesley Publishing Company. Inc, 1989
- [6] Small Industrial Robot System Technical Manual, A460 series, CRS Plus inc., 1990