Motor Jig 프로젝트

임베디드스쿨1기
Lv1과정
2021. 04. 02
박성환

0. Overview

- **개발컨셉**
  - 흔히 사용되는 32bit MCU를 기반으로 하여 시중의 다양한 모터 및 드라이버 테스트가 가능한 지그 제작
  - 제작한 지그를 활용하여 모터 드라이버에서 정한 프로토콜에 따라 스텝모터를 제어
  - 완성된 지그 및 드라이버를 PC에서 구동시킬 수 있는 간단한 GUI 형태 프로그램 구성

- **기대효과**
  - 국내에서 많이 사용하는 MCU 사용에 익숙해짐
  - 단순한 아스키명령에서 바이너리 방식의 데이터 송수신 개발을 해봄으로써 패킷방식의 데이터 송수신 개발 방법에 대해 이해
  - C언어 활용 (void 포인터,함수포인터 등)에 능숙해짐
  - 다양한 모터들을 붙여 테스트 할 수 있는 나만의 지그가 생김
  - 간단한 GUI 프로그램을 개발해 봄으로써 쉽게 테스트에 필요한 환경 구성이 가능해짐

- **필요부품**
  - Ezistep모터드라이버/모터Set x 1
  - SMPS 24V two out x 1
  - Cable류 x 5
  - RS485 변환 보드 x1
  - Control 보드 x 1

- **사용환경**
  - MCU : stm32H7 (cortex-m7)
  - SDK : stm32ide(제조사 제공), Visual Studio2017
  - 개발언어 : C/C#

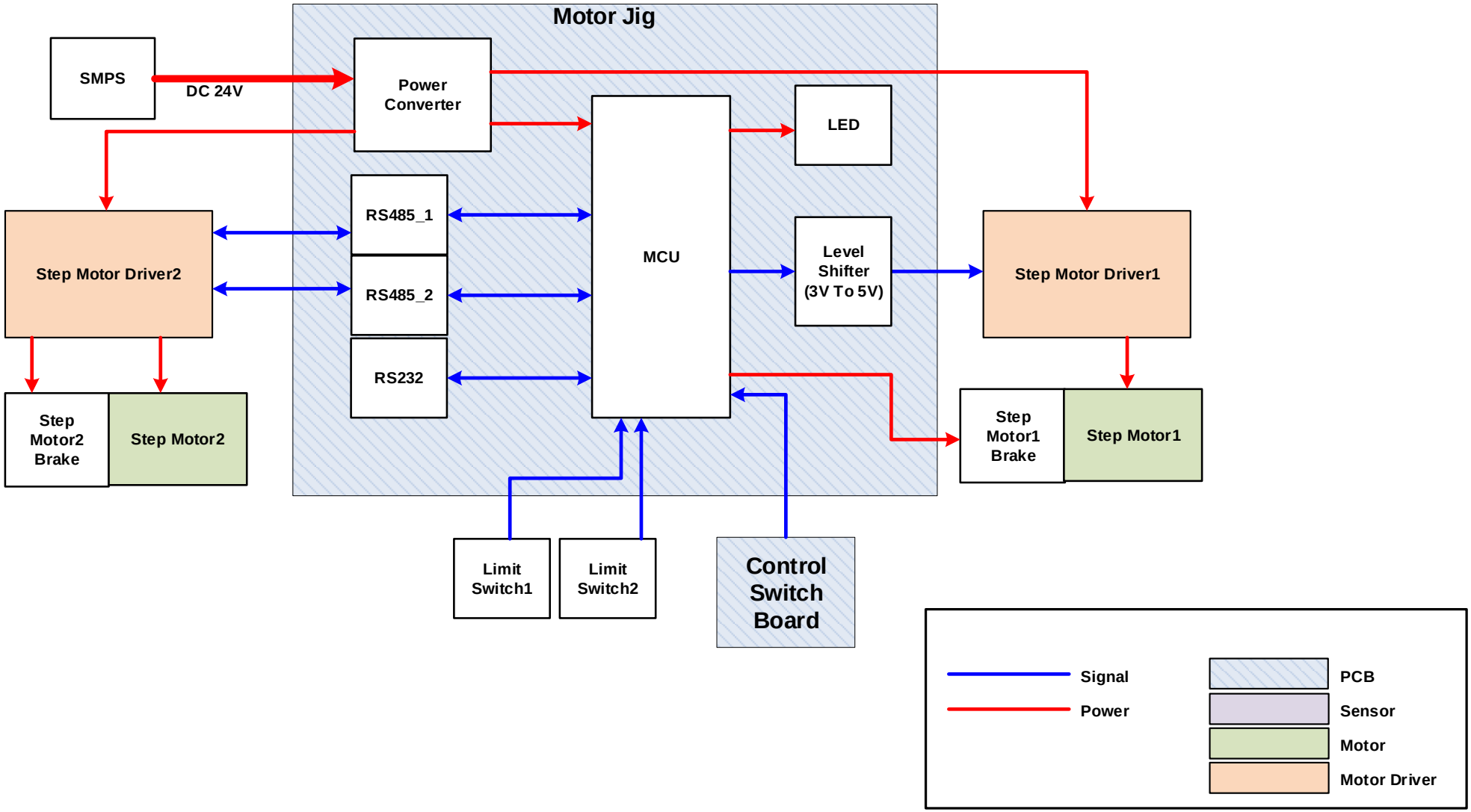
0. Overview

- **개발 일정**

주요 업무	11월	12월	1월	2월	3월
컨셉 및 사양 결정	→				
회로 설계		→			
Artwork			→		
PCB & SMT			→		
FW				→	→
SW					→
발표 준비					→

1. HW

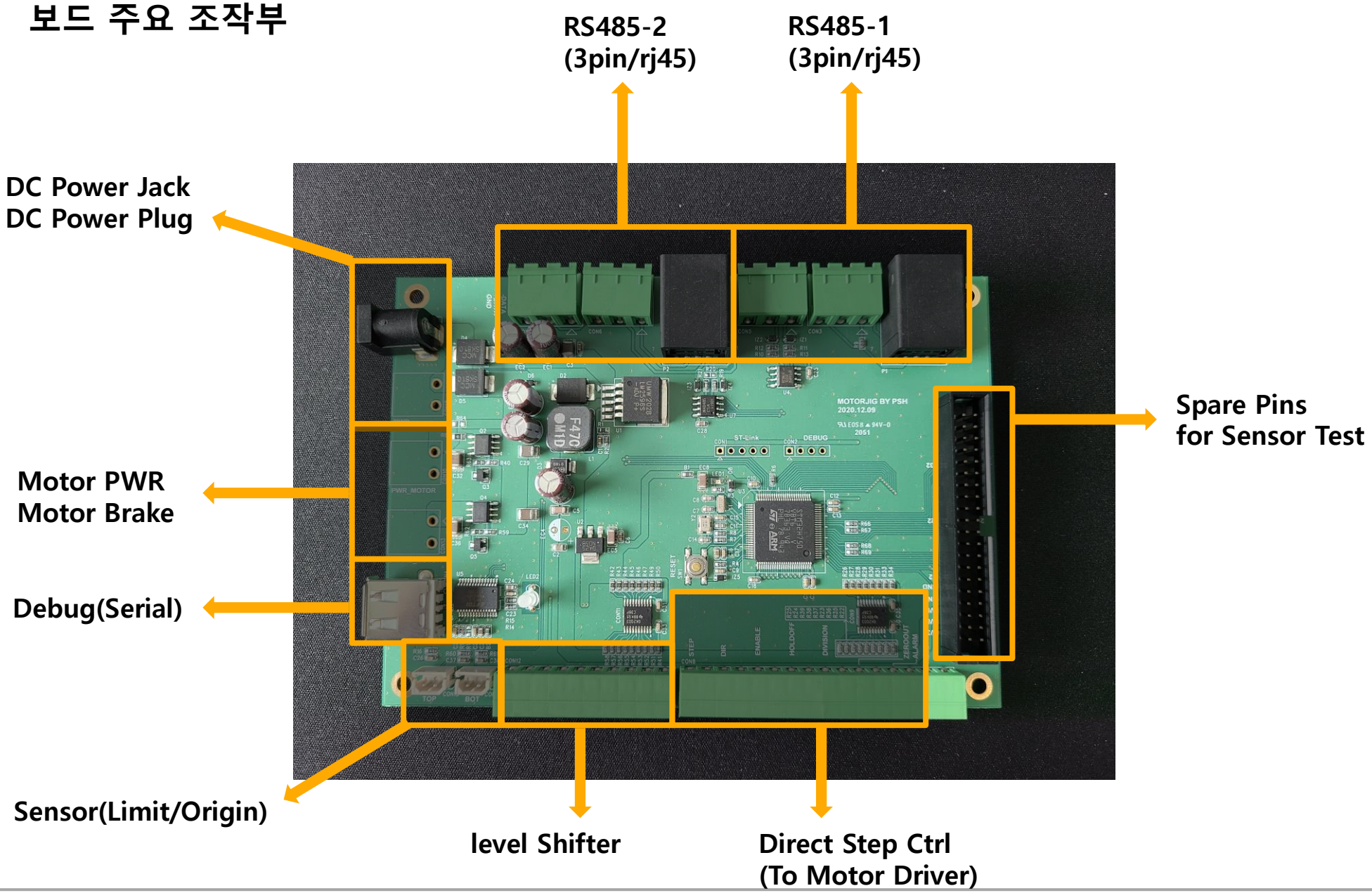
- **HW 블록 다이어그램(Plan)**



포기하면 얻는 건 아무것도 없다.

1. HW

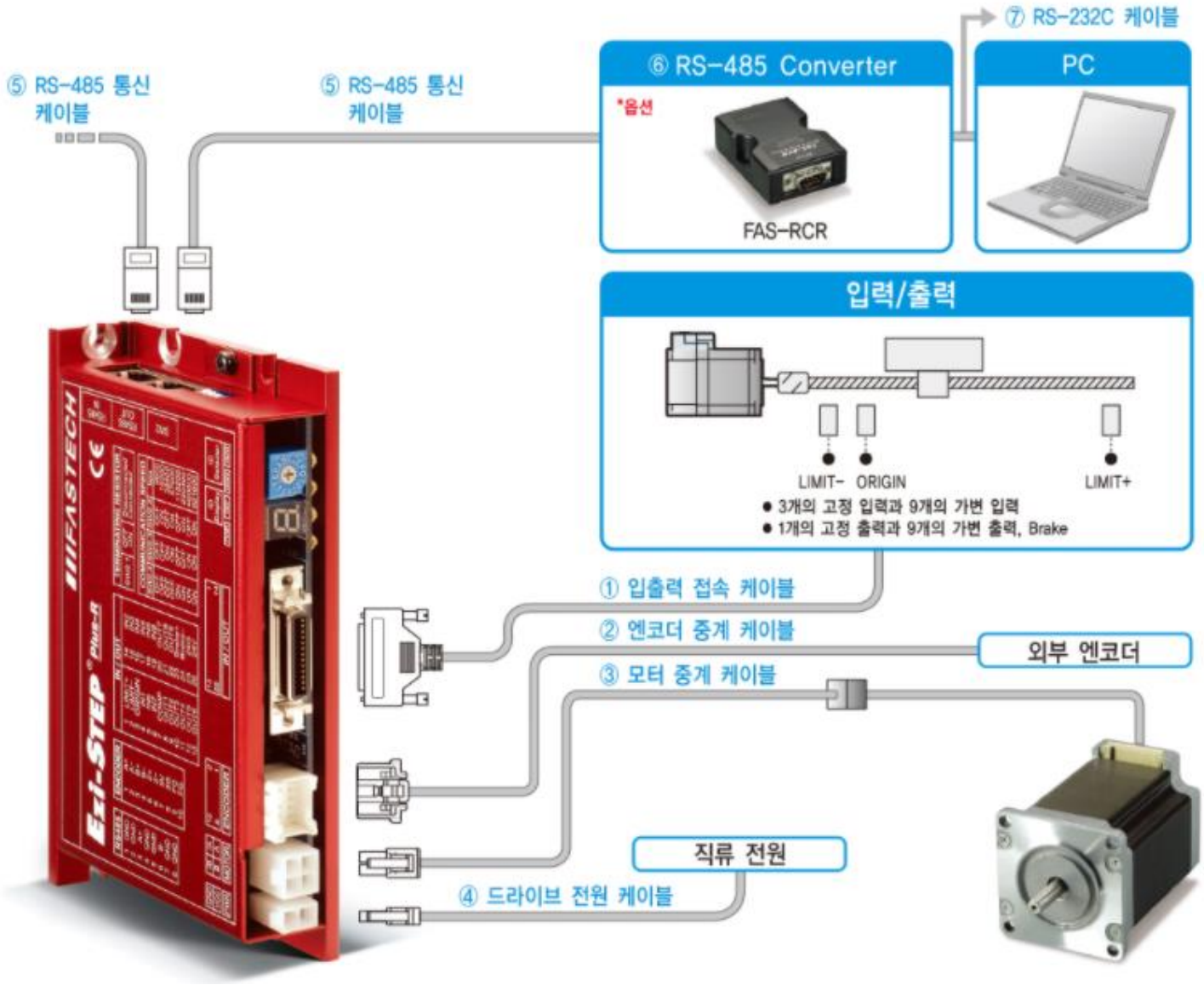
- **보드 주요 조작부**



포기하면 얻는 건 아무것도 없다.

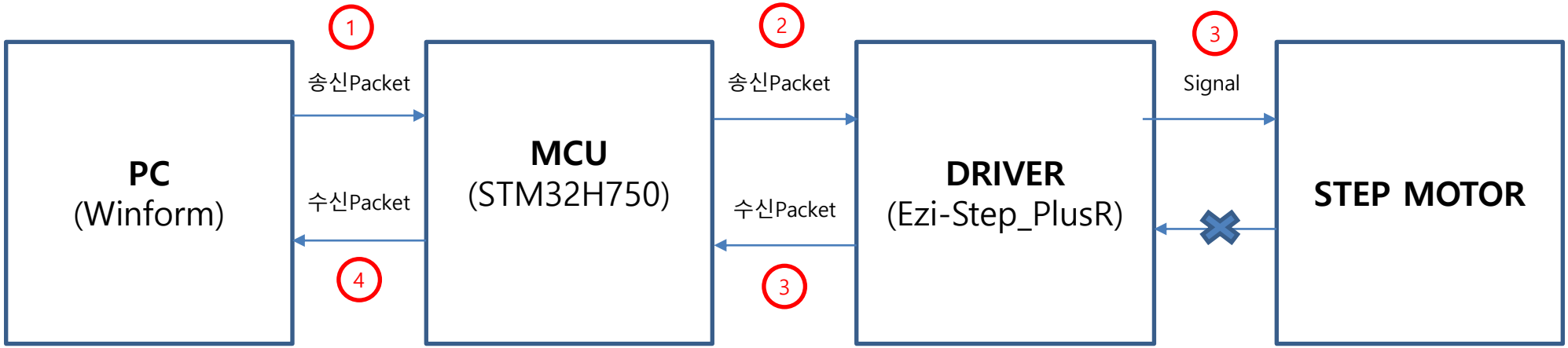
1. HW

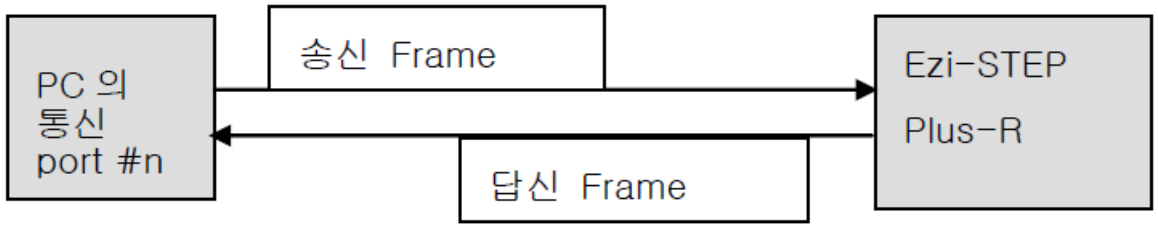
- **시스템 구성도**


⑦ RS-232C 케이블
⑥ RS-485 Converter
*옵션
FAS-RCR
PC
⑤ RS-485 통신 케이블
⑤ RS-485 통신 케이블
입력/출력
LIMIT- ORIGIN LIMIT+
● 3개의 고정 입력과 9개의 가변 입력
● 1개의 고정 출력과 9개의 가변 출력, Brake
① 입출력 접속 케이블
② 엔코더 중계 케이블
③ 모터 중계 케이블
외부 엔코더
직류 전원
④ 드라이브 전원 케이블
MIFASTECH
Ez-STEP

2. FW

- **통신 전체 구조**





포기하면 얻는 건 아무것도 없다.

2. FW

- **FRAME 기본 송신 구조**

Header	*Frame Data*	Tail
0xAA 0xCC	4~252 bytes	0xAA 0xEE

Slave ID	Frame type	Data	CRC	
1 byte	1 bytes	0 ~ 248 bytes.	2 bytes	
			Low byte	High byte

1) 0xAA 0xCC : Frame의 시작 Header
2) 0xAA 0xEE : Frame의 종료 Tail
3) Slave ID : PC 통신의 모듈과 연결된 번호
4) Frame Type : 해당 Frame의 명령어 종류를 지정
5) Data : 이 항의 데이터 구조 및 길이 등은 Frame Type에 따라 정해짐
6) CRC : CRC16 계산 방식 중 2가지 ('0xA001', CRC-16-IBM) 사용

- **FRAME 기본 수신 구조**

Header	*Frame Data*	Tail
0xAA 0xCC	4~252 bytes	0xAA 0xEE

Slave ID	Frame type	Data		CRC	
1 byte	1 bytes	1 bytes.	0 ~ 247bytes.	2 bytes	
		통신 상태	답신 데이터	Low byte	High byte

1) 송신 구조와의 차이점은 '통신 상태 ' 가 추가되었다는 점

2. FW

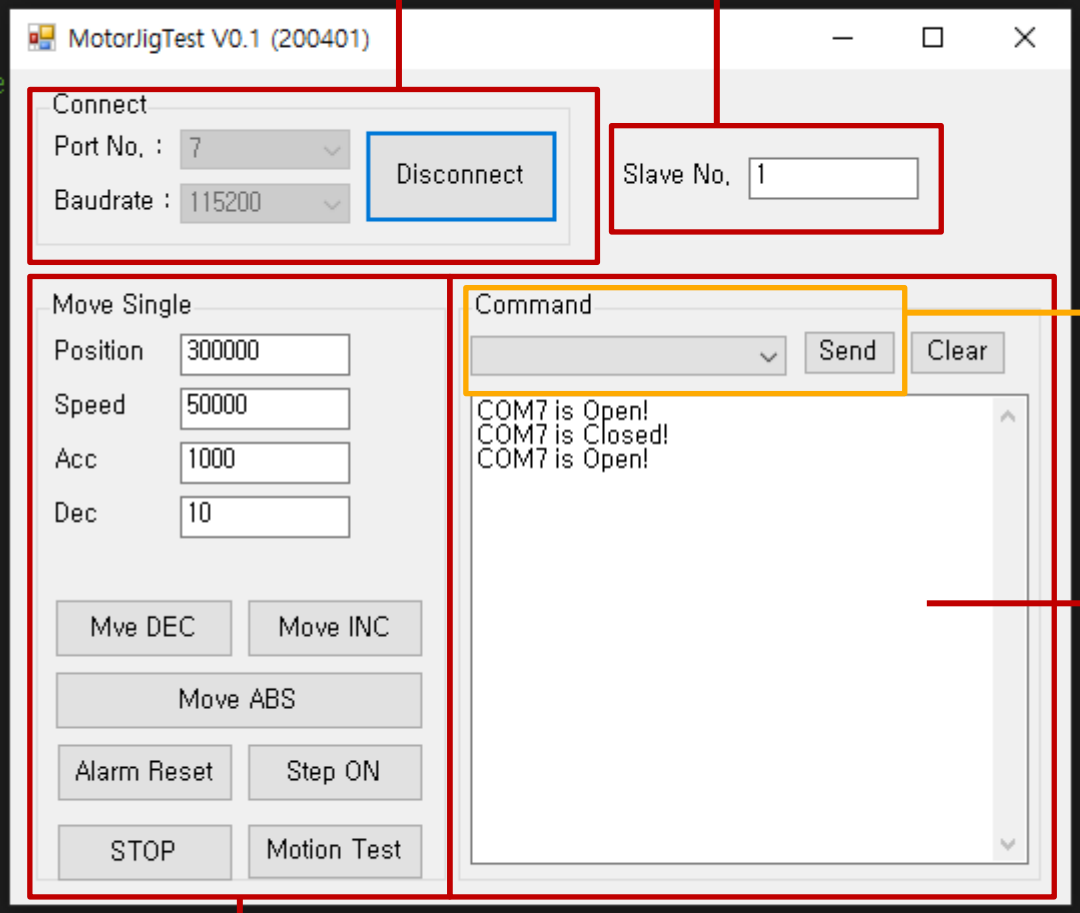
- **Frame 송수신 명령 구성**

Frame type	라이브러리 명	내용
0x01 (1)	FAS_ GetSlaveInfo	연결된 slave의 종류와 프로그램 version 정보를 의뢰함. 송신 : 0 byte 답신 : 1~248 bytes 1 byte: 통신 상태 / 1 bytes: Slave 종류 / 0~246 bytes: ACII string with NULL byte (strlen() + 1 byes) ♦ Slave 종류 : 20 : Ezi-STEP Plus-R ST 60 : Ezi-STEP Plus-R MINI 1 : Ezi-SERVO Plus-R ST
0x05 (5)	FAS_ GetMotorInfo	Slave에 연결된 모터의 종류에 대한 정보를 의뢰함. 송신 : 0 byte 답신 : 1~246 bytes 1 byte: 통신 상태 / 1 bytes: Motor 번호 (1~255) / 0~246 bytes: ACII string with NULL byte (strlen() + 1 byes) ♦ Motor 종류 : 「1-2-7. 모터 종류」의 표를 참조.
0x10 (16)	FAS_ SaveAllParameters	현재 설정된 파라미터 값과 입출력 신호의 할당값들을 드라이브의 ROM 메모리에 저장함. 이는 드라이브 전원을 OFF 해도 값들이 저장되도록 한다. 따라서 'FAS_SetParameter' 와 'FAS_SetIOAssignMap' 에서 설정된 값들이 함께 저장됨. 송신 : 0 byte 답신 : 1 byte 1 byte: 통신 상태
0x11 (17)	FAS_ GetRomParameter	ROM 메모리 영역의 특정 파라미터값을 읽어들임. 송신 : 1 byte 1 byte: 파라미터 번호 (0~28) 답신 : 5 bytes 1 byte: 통신 상태 / 4 bytes: 파라미터 값 「1-2-2. 파라미터 목록표」 를 참조.
0x12 (18)	FAS_ SetParameter	특정 파라미터 값을 드라이브의 RAM 메모리에 저장함. 송신 : 5 bytes 1 byte: 파라미터 번호 (0~28) / 4 bytes: 파라미터 값 답신 : 1 byte 1 byte: 통신 상태 「1-2-2. 파라미터 목록표」 를 참조.
0x13 (19)	FAS_ GetParameter	RAM 메모리 영역의 특정 파라미터값을 읽어들임. 송신 : 1 byte 1 byte: 파라미터 번호 (0~28) 답신 : 5 bytes 1 byte: 통신 상태 / 4 bytes: 파라미터 값 「1-2-2. 파라미터 목록표」 를 참조.
0x20 (32)	FAS_ SetIOOutput	제어 출력단의 출력 신호 레벨을 설정함. 송신 : 8 bytes 4 bytes: I/O set mask 값 / 4 bytes: I/O clear mask 값 Set mask의 특정 bit값이 1이면 해당 출력단 신호는 [ON]이 된다. Clear mask의 특정 bit값이 1이면 해당 출력단 신호는 [OFF]가 된다. 자세한 사항은 「1-2-3. 제어출력핀의 bit 설정」항을 참조. 답신 : 1 byte 1 byte: 통신 상태
0x21 (33)	FAS_ SetIOInput	제어 입력단의 입력 신호 레벨을 설정함. 송신 : 8 bytes 4 bytes: I/O set mask 값 / 4 bytes: I/O clear mask 값 Set mask의 특정 bit값이 1이면 해당 입력단 신호는 [ON]이 된다. Clear mask의 특정 bit값이 1이면 해당 입력단 신호는 [OFF]가 된다. 자세한 사항은 「1-2-4. 제어입력핀의 bit 설정」항을 참조. 답신 : 1 byte 1 byte: 통신 상태

3. SW

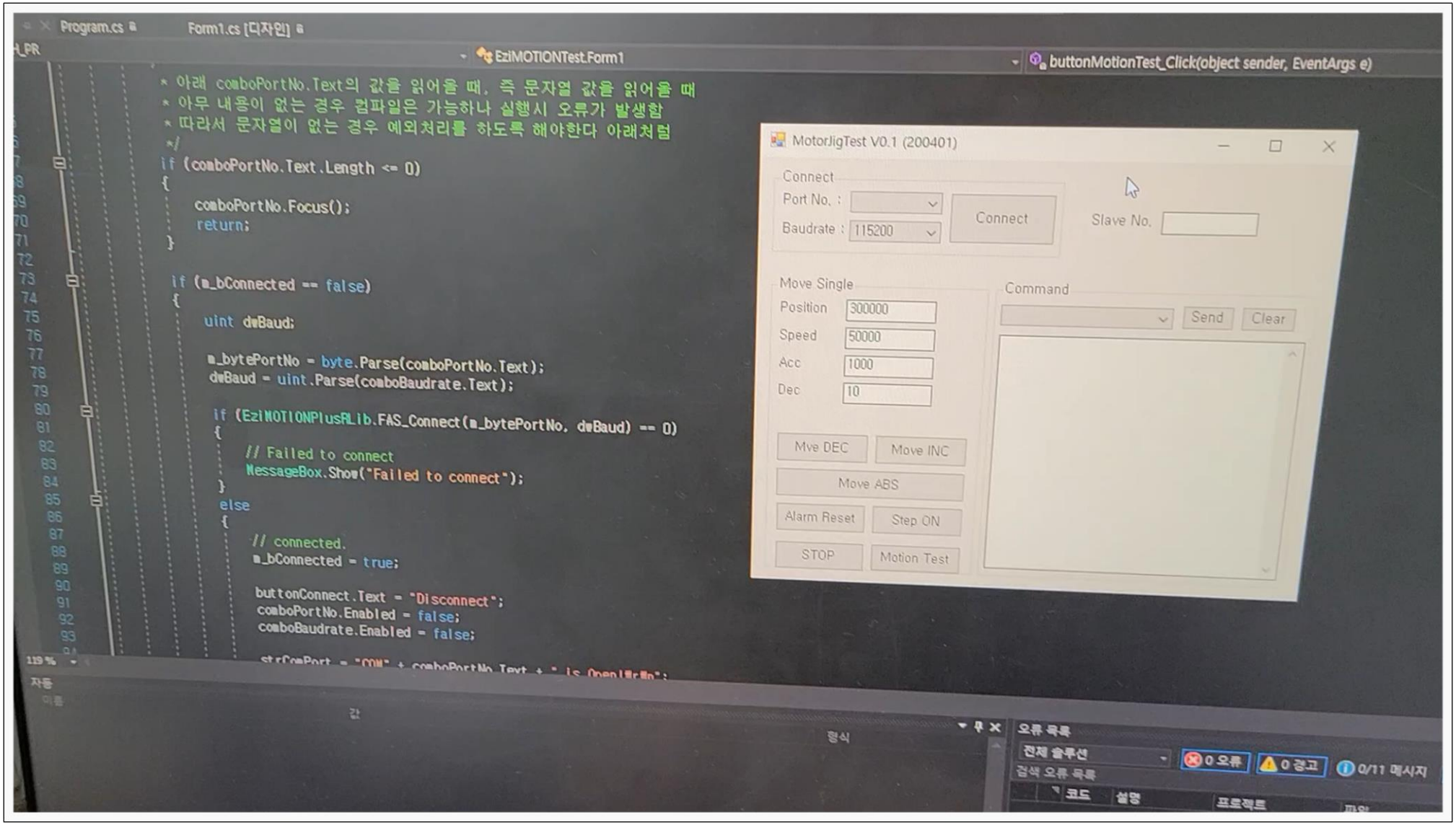
- **제조사 제공 DLL을 사용하여 작성**

```
참조 1개 | 0번 변경 | 만든 이 0명, 변경 내용 0개
private void Form1_Load(object sender, EventArgs e)
{
    UpdateSerialPortList();
    comboBaudrate.SelectedIndex = 4;    // default baudrate

    textPosition.Text = "300000";
    textSpeed.Text = "50000";
    textAccelTime.Text = "1000";
    textDecelTime.Text = "10";
}

참조 1개 | 0번 변경 | 만든 이 0명, 변경 내용 0개
private void UpdateSerialPortList()
{
    comboPortNo.Items.Clear();

    // Port No.
    string[] portlist = SerialPort.GetPortNames();

    List<int> PortNoList = new List<int>();

    foreach (string port in portlist)
        PortNoList.Add(int.Parse(port.Substring(3)));

    PortNoList.Sort();

    foreach (int portno in PortNoList)
        comboPortNo.Items.Add(string.Format("{0}", portno));
```



포기하면 얻는 건 아무것도 없다.

4. 시연영상



포기하면 얻는 건 아무것도 없다.

5. 이슈

- **주요 이슈사항 정리**

이슈	원인	해결
PC프로그램으로부터 MCU(=slave1)가 데이터를 받은 후, MCU가 받은 데이터를 모터 드라이버(=slave2)에 보내는데 보낸 데이터가 PC 프로그램에 읽혀 알 수 없는 정보를 표시함	RS485 통신으로 모든 slave 기기의 RX는 RX끼리 TX는 TX끼리 묶여있음 2선 반이중 통신 방식인데 송수신이 겹치지 않는 선에서는 데이터를 원만하게 받아야하는데 PC 프로그램에서 잘못 받았을시 처리가 잘 안되어 있음	임시로 RS485 포트를 추가로 구성하여 송수신 포트를 다르게 가져감
송수신 데이터 개수를 제공하지 않고 사용자가 맵핑해서 사용해야 하는 상황 발생	기본적으로 매뉴얼에서 제공하는 송수신 개수를 직접 코드에 적용하여 구현함	해결은 하였으나 일부 답신에서 데이터 개수가 정해지지 않고 가변할 경우는 처리하지 못함
송수신 데이터에 한 번 잘못 받은 경우 이후 보드에서 지속적으로 송수신을 못하는 문제 발생	각각의 패킷 바이트마다 Case문으로 처리를 하는데 잘못들어오는 경우에 대한 예외처리가 제대로 되어 있지 않아 한 번 잘 못받으면 계속 오류 Case문에서 빠져나오지 못함	각 Case문에서 예상되는 예외처리 코드 추가하여 해결
dll 사용하여 해당 포트/슬레이브에 명령을 주면 응답이 전부 에러 발생	메인보드를 Slave '1'로 설정하였는데 메인보드는 slave 존재여부에 대해 알려주는 응답 함수가 따로 없기에 에러가 발생함	메인보드는 디폴트로 두고 드라이버를 찾는 코드 구현
수신 데이터가 정확히 정해지지 않고 가변되는 경우 제대로 수신 받지 못하는 문제 발생	수신 패킷 데이터중 데이터가 정해지지 않고 가변되는 경우에 대한 처리 예외코드가 따로 존재 x	예외처리 코드로 구현함으로써 해당 문제 해결

4. 이슈

- **주요 이슈사항 정리**

이슈	원인	해결
PC프로그램으로부터 MCU(=slave1)가 데이터를 받은 후, MCU가 받은 데이터를 모터 드라이버(=slave2)에 보내는데 보낸 데이터가 PC 프로그램에 읽혀 알 수 없는 정보를 표시함	RS485 통신으로 모든 slave 기기의 RX는 RX끼리 TX는 TX끼리 묶여있음 2선 반이중 통신 방식인데 송수신이 겹치지 않는 선에서는 데이터를 원만하게 받아야하는데 PC 프로그램에서 잘못 받았을시 처리가 잘 안되어 있음	임시로 RS485 포트를 추가로 구성하여 송수신 포트를 다르게 가져감
송수신 데이터 개수를 제공하지 않고 사용자가 맵핑해서 사용해야 하는 상황 발생	기본적으로 매뉴얼에서 제공하는 송수신 개수를 직접 코드에 적용하여 구현함	해결은 하였으나 일부 답신에서 데이터 개수가 정해지지 않고 가변할 경우는 처리하지 못함
송수신 데이터에 한 번 잘못 받은 경우 이후 보드에서 지속적으로 송수신을 못하는 문제 발생	각각의 패킷 바이트마다 Case문으로 처리를 하는데 잘못들어오는 경우에 대한 예외처리가 제대로 되어 있지 않아 한 번 잘 못받으면 계속 오류 Case문에서 빠져나오지 못함	각 Case문에서 예상되는 예외처리 코드 추가하여 해결
dll 사용하여 해당 포트/슬레이브에 명령을 주면 응답이 전부 에러 발생	메인보드를 Slave '1'로 설정하였는데 메인보드는 slave 존재여부에 대해 알려주는 응답 함수가 따로 없기에 에러가 발생함	메인보드는 디폴트로 두고 드라이버를 찾는 코드 구현
수신 데이터가 정확히 정해지지 않고 가변되는 경우 제대로 수신 받지 못하는 문제 발생	수신 패킷 데이터중 데이터가 정해지지 않고 가변되는 경우에 대한 처리 예외코드가 따로 존재 x	예외처리 코드로 구현함으로써 해당 문제 해결

감사합니다.