Master(PLC)와 Slave(Motor)의 프로토콜

1. PLC는 Master 역할이고 Motor는 Slave 역할을 합니다.

2. Master(PLC)가 Slave(Motor)로 데이터 전송하는 방법은 RS485 통신 입니다.

- BaudRate는 현재 9600bps(default) 입니다.
  9600, 14400, 19200, 28800, 38400, 57600, 76800, 115200, 230400, 250000 등이 가능합니다.
- 데이터 전송과 수신은 "Hand Shake" 방식 입니다. Master(PLC)가 명령 데이터를 보내면 모터는 응답 데이터를 보냅니다. 즉, 9바이트 데이터를 전송하면 모터는 100[us](시간 간격은 조정 가능) 이내에 응답 데이터 9바이트를 보냅니다.

3. Master(PLC)가 Slave(Motor)로 데이터 전송
   - 그림 1은 Master(PLC)가 Slave(Motor)로 데이터를 전송하는 데이터 포맷 입니다. 9Byte로 구성됩니다.

Bytes	Meaning
1	Module address
1	Command number
1	Type number
1	Motor or Bank number
4	Value (MSB first!)
1	Checksum

그림 1 호스트에서 모터로 전송되는 데이터 포맷

Bytes	Meaning
1	Reply address
1	Module address
1	Status (e.g. 100 means "no error")
1	Command number
4	Value (MSB first!)
1	Checksum

그림 2 호스트가 모터로부터 수신되는 데이터 포맷

- 여기서 Checksum 계산은 다음과 같습니다.(C언어로 표현)

```
unsigned char i, checksum;
unsigned char command[9];

checksum = command[0];
for(i = 1; i < 8; i++)
{
    checksum += command[i];
}
command[8] = checksum;
```

- 위와 같이 command[0]부터 순서대로 command[8]까지 전송하면 됩니다.
- 그림 2는 Master(PLC)가 Slave(Motor)로 데이터를 수신하는 데이터 포맷 입니다. 9Byte로 구성됩니다

- 그림3은 그림2의 3번째 바이트 "Status code"에 대한 것 입니다.

Code	Meaning
100	Successfully executed, no error
101	Command loaded into TMCL program EEPROM
1	Wrong checksum
2	Invalid command
3	Wrong type
4	Invalid value
5	Configuration EEPROM locked
6	Command not available

그림 1 Status code

4. Master(PLC)가 Slave(Motor)에게 "생산제품에 대한 기본 설정 값" 송신 및 수신
   - 아래 표는 모터 내부메모리 번지와 "예제 값"에 대한 전송 데이터를 예시 합니다. 모터 번호는 "0x01" 입니다.
   - 여기서 "송신"은 Master(PLC)가 Slave(Motor)에게 9바이트 데이터 전송이고 "수신"은 수신된 데이터 입니다.

설명	내부메모리 번지		예제 값(4byte)	Master(PLC)가 Slave(Motor)로 데이터 전송	
제품의 텝수	9	9 (0x09)	12 (0x0000000c)	송신	0x01 0x09 0x09 0x02 0x00 0x00 0x00 0x0c 0x21
				수신	0x01 0x01 0x64 0x09 0x00 0x00 0x00 0x0c 0x7b
삽입위치	10	10 (0x0a)	12345 (0x00003039)	송신	0x01 0x09 0x0a 0x02 0x00 0x00 0x30 0x39 0x7f
				수신	0x01 0x01 0x64 0x09 0x00 0x00 0x30 0x39 0xd8
1텝 이동위치	11	11 (0x0b)	1600 (0x00000640)	송신	0x01 0x09 0x0b 0x02 0x00 0x00 0x06 0x40 0x5d
				수신	0x01 0x01 0x64 0x09 0x00 0x00 0x06 0x40 0xb5
2텝 이동위치	12	12 (0x0c)	1900 (0x0000076c)	송신	0x01 0x09 0x0c 0x02 0x00 0x00 0x07 0x6c 0x8b
				수신	0x01 0x01 0x64 0x09 0x00 0x00 0x07 0x6c 0xe2
…	…	…	…		…
n텝 이동위치	10+n	22 (0x16)	-1850 (0xfffff8c6)	송신	0x01 0x09 0x16 0x02 0xff 0xff 0xf8 0xc6 0xde
				수신	0x01 0x01 0x64 0x09 0x0ff 0xff 0xf8 0xc6 0x2b
텝이동 종료 의미	10+n+1	23 (0x17)	0 (0x00000000)	송신	0x01 0x09 0x17 0x02 0x00 0x00 0x00 0x00 0x23
				수신	0x01 0x01 0x64 0x09 0x00 0x00 0x00 0x00 0x6f

5. Master(PLC)와 Slave(Motor)의 State

- 아래 표는 각 state에서 Master(PLC)와 Slave(Motor)의 역할을 정의한 것 입니다.

state	Master(PLC)	Slave(Motor)	비고
S1	-제품 준비하는 상태	-대기 상태	
S2	-mOUT1를 HIGH	-삽입위치로 이동 -sOUT1를 HIGH	
S3	-제품 삽입하기 -mOUT2를 HIGH		
S4		-1번텝 위치로 이동 -sOUT2를 HIGH	
S5	-mOUT2를 LOW -제품 cutting 구간(1번텝) -mOUT2를 HIGH	-sOUT2를 LOW	
S6		-2번텝 위치로 이동 -sOUT2를 HIGH	
S7	-mOUT2를 LOW -제품 cutting 구간(2번텝) -mOUT2를 HIGH	-sOUT2를 LOW	
	…	…	
S8		-n번텝 위치로 이동 -sOUT2를 HIGH	
S9	-mOUT2를 LOW -제품 cutting 구간(n번텝) -mOUT2를 HIGH	-sOUT2를 LOW	
S10	-제품 들어내기 -mOUT1과 mOUT2를 LOW	-HOME이동 -sOUT1를 LOW	

- 그림 4는 Master(PLC)와 Slave(Motor)의 state에 따른 I/O signal에 대한 것 입니다.

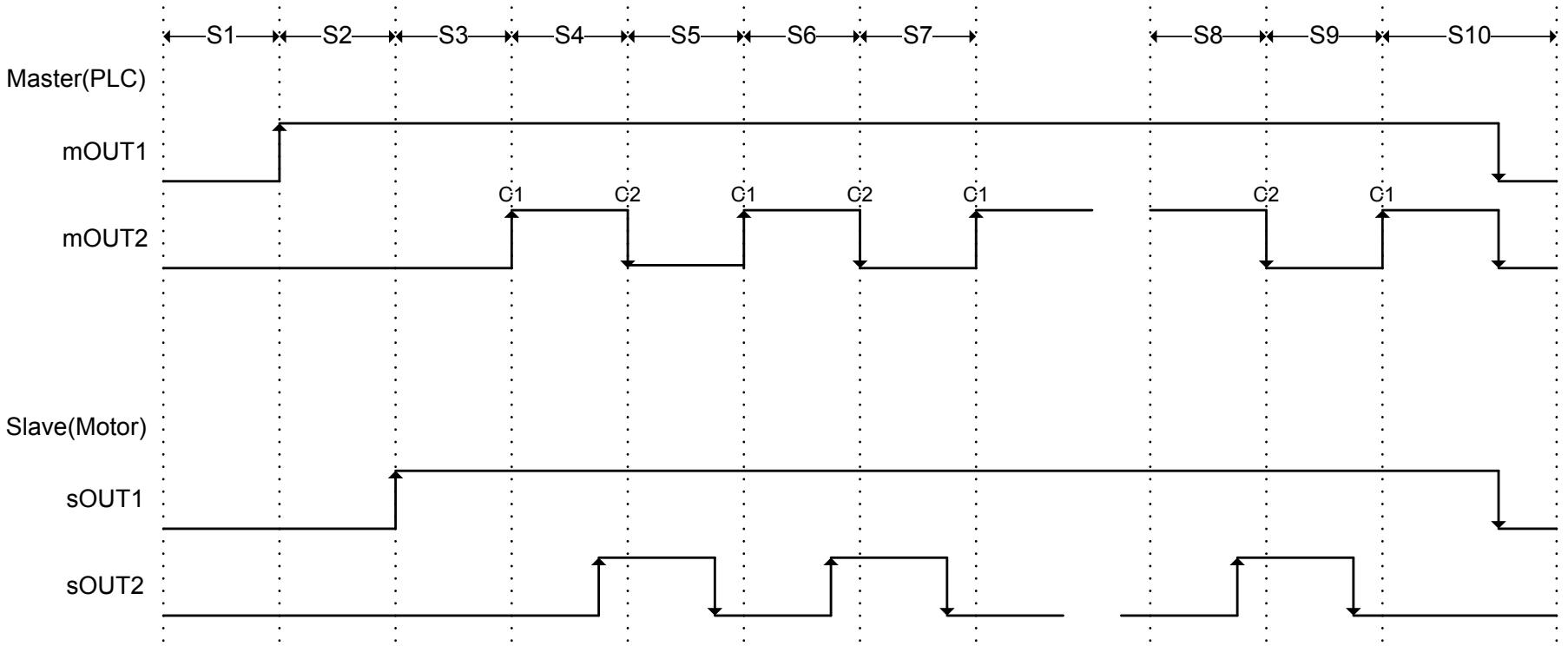


그림 4 Master(PLC)와 Slave(Motor)의 state와 I/O signal

- 그림5는 I/O 포트 연결(왼쪽)에 대한 것과 ARM 운동(오른쪽)에 대한 것 입니다.

I/O 포트 연결	
Master(PLC)	Slave(Motor)
mOUT1	sIN1
mOUT2	sIN2
mIN1	sOUT1
mIN2	sOUT2

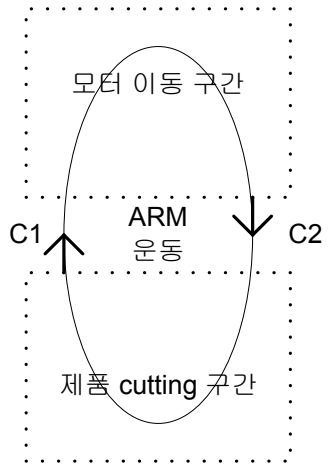


그림 5 I/O 포트 연결(왼쪽)과 ARM 운동(오른쪽)