LED Power Controller

CVT-Emini Series Protocol Manual



CAS
VISION TECH

목차

1. 시작 하기
    1. 문서 리비전 정보

2. 프로토콜 사양
    1. 프로토콜 사양
        2.1.1
    2. 통신 명령
        1. 제품 초기화
        2. 밝기 제어
        3. ON/OFF 제어
        4. 밝기 저장 및 불러오기
        5. 에러
    3. 패킷 통신

1. 시작 하기

Emini Series Protocol은 ERI의 고유 자산입니다.

해당 문서의 재배포 및 내용 수정은 불가합니다.

본 문서의 내용은 제품 성능의 향상을 위해 수정 및 변경이 일어날 수 있습니다.
또한 본 프로토콜을 사용하는 Emini 시리즈 및 관련 제어기의 제품에서도 성능 향상을 위한 수정 및 변경이 일어날 수 있습니다.

1.1 문서 리비전 정보

날짜	내용	비고
2011. 07. 10.	프로토콜 출시	Ver0.5
2012. 04. 03.	Emini Type Protocol 추가	Ver1.0

2. 프로토콜 사양

2.1 프로토콜 사양

2.1.1 통신 포맷

모드 : RS232/USB 통신

보드레이트 : 38,400 bps

데이터 비트 : 8 bit

패러티 비트 : 없음

스톱 비트 : 1 bit

흐름 제어 : 없음

2.2 통신 명령

* Note : 전면 패널의 Mode가 Remote일때 동작되고 , Manual 일 경우 동작을 하지 않습니다

전송되는 모든 값은 아스키코드 값입니다.

2.2.1 제품 초기화

"reset\r"

-. 모든 동작을 전원이 켜졌을 경우의 상태로 되돌림.

-. 제품의 모델명, 각 채널의 상태등 제어기의 정보를 얻을 수 있음.

2.2.2 밝기 제어

"setbrightness [ch] [bright]\r"

- 선택 된 채널의 밝기 값을 조정.

- [ch] : 조명의 채널을 선택한다. (4채널 : 0~3) (8채널 : 0~7)

- [bright] : 조명의 밝기값 (0~255)

"# [ch] [bright]\r"

-."setbrightness"와 동일한 기능을 함.

"getbrightness [ch]\r"

- 선택 된 채널의 밝기 값을 불러 옴.
- [ch] : 밝기값을 읽을 채널 (4채널 : 0~3) (8채널 : 0~7)
- return : "getbrightness [index] [ch] [bright]\n"

"$ [ch]\r"

- getbrightness와 동일한 기능을 함.
- return : "$ [index] [ch] [bright]\n"

2.2.3 ON/OFF 제어

"seton [ch] [on/off]\r"

- 특정 채널 하나의 On/Off를 결정 함.
- [ch] : 조명의 채널을 선택한다. (4채널 : 0~3) (8채널 : 0~7)
- [on/off] : 0 - Off , 1 - On

"setonex [on/off]\r"

- 채널들 혹은 모든 채널의 On/Off를 결정 함.
- [on/off] hex number로 표현하여 전체 채널을 On/Off 할 수 있음.
  ex) "setonex 0F\r" , 0~3번 채널을 On
  ex) "setonex 01\r" , 0번채널(채널1) On , 1~3번채널 Off

"geton [ch]\r"

- 특정 채널 하나의 On/Off 상태를 불러 옴.
- [ch] : 조명의 채널을 선택한다. (4채널 : 0~3) (8채널 : 0~7)
- return : "geton [index] [ch] [on/off]\n"

"getonex\r"

- 채널들의 On/Off 상태를 불러 옴.
- return : "getonex [index] [on/off]\n" ex) "getonex 0 0F\n"

2.2.4 밝기 저장 및 불러오기

"savebright [index]\r"

- 모든 채널의 현재 정보(밝기 값, On/Off 상태)를 저장 함.
- [index] : 저장될 메모리 인덱스를 선택한다. (0~9)
- 현재 설정된 밝기값 및 On/Off 상태를 해당되는 인덱스에 저장한다.

"loadbright [index]\r"

- 모든 채널의 현재 정보(밝기 값, On/Off 상태)를 불러 옴.

  불러 온 값을 각 채널에 모두 적용 됨.

- [index] : 로드할 메모리 인덱스를 선택한다. (0~9)

- 선택한 인덱스의 밝기값과 On/Off상태를 적용한다.

"getbrightindex\r"

- 모든 채널의 현재 정보(밝기 값, On/Off 상태)를 불러 옴.

  불러 온 값을 각 채널에 모두 적용 됨.

- 현재 load한 인덱스 값을 return한다.

- return : "getbrightindex [index]\n"

* 밝기값은 밝기제어 명령 참조

2.2.5 에러

"geterrstatus\r"

- 현재의 에러상태를 return한다. 또는 에러발생시 1초단위로 자동으로 report함.

- return : "geterrstatus [err] [code]\n"

- [err] : 0 - none error , 1 - error (hex number)

- [code] : error code (bit단위로 에러상태 체크) (hex number)

  0x02 : 과온도 보호 상태

  0x10 : 과전류 보호 상태 (0~3번채널)

  0x20 : 과전류 보호 상태 (4~7번채널)

  0x40 : 과전류 보호 상태 (8~11번채널)

  0x80 : 과전류 보호 상태 (12~15번채널)

  ex) "geterrstatus 1 12\n"

  과온도이고, 0~3번채널중 과전류 보호 상태이다.

2.3 패킷 통신

//채널이 많을경우 아래의 방식으로 모든 채널의 밝기제어를 한번에 할 수 있음.

//console command에서 별도의 셋팅이 필요없이 자동인식으로 동작 됨.

```
#define ERI_PACKET_CODE_HEADER          0xFE
#define ERI_PACKET_CODE_END_FRAME     0xFD
#define ERI_PACKET_CODE_BRIGHT          0x06
typedef struct{
   unsigned char header[2];              //0xFE,0xFE
   unsigned char cmd;                    //0x06
   unsigned char packet_length[2];    //header,cmd,packet_length,end를 제외한 길이(32)
                                         //packet_length[0] = MSB(0),packet_length[1]=LSB (3
   unsigned char bright[16][2];          //bright[ch][0]=MSB,bright[ch][1]=LSB
   unsigned char end;//0xFD
} ERI_PACKET_T;
```