

GRAND MA 2

실전활용법 - 4

- EXECUTOR를 활용하여 CUE 만들기, 창 구성하기



글.
강우빈 KB아트홀 조명감독

연재 목차 : GRAND MA 2 실전활용법(기본편)

- Part 1.외관, 하드웨어 구성, 콘솔과 부속기기들
- Part 2.조명기 Patch, 기본적인 Preset 구성하기 ①, ②
- Part 3. EXECUTOR를 활용하여 CUE 만들기, 창 구성하기
- Part 4.CUE 편집하기, MIB 기본활용
- Part 5.EFFECT, LAYOUT 기본활용
- Part 6.나만의 EFFECT 만들기

들어가며

지금까지 콘솔을 처음 접하고, 메모리에 필요한 기본요소들을 간단하게 구성하였습니다. 기존의 방법보다 매우 빠르고 효율적으로 구성할 수 있었을 것입니다. 이제는 이를 바탕으로 기초적인 메모리에 들어가 보도록 하겠습니다.

이번 편에서는 EXECUTOR를 활용하여 CUE 만들기, 창 구성하기에 대해서 다루겠습니다.

이번 편은 다음과 같은 순서로 이루어집니다.

1. 메모리 방식 정하기(메모리방식, Store option 정하기)
2. Preset 창 구성하기
3. 구성된 Preset을 이용하여 Executor CUE 메모리

GRAND MA 2의 메모리 방식에 대해

앞서 첫 편에서 GRAND MA 2의 가장 큰 특징으로 하나의 시퀀스(움직임)를 만들 때 여러 가지 갈래의 길이 있다고 언급하였습니다. 메모리도 마찬가지로 여러 가지의 길이 있습니다. 종급 이상으로 갈수록 어떤 길을 선택하느냐에 따라 메모리가 매우 효율적이고 속도 면에서 차이가 많이 날 수 있습니다. 이번 편에서는 큰 카테고리의 길을 소개하겠습니다.

Program Memory

Pool 창에 입력하는 모든 과정, Encoder를 돌려 원하는 값을 만드는 것, Color Picker를 찍어서 컬러를 바꾸는 모든 과정을 Program Memory라 명할 수 있습니다. 이 경우에 조명 메모리값은 Active 된 값으로 존재합니다. 콘솔 수치상에는 붉은색으로 입력됩니다. 이 경우에 만든 값들이 별도로 저장된 것은 아닙니다. 단지 불러올 뿐입니다. Clear 키를 누르면서 Active 상태를 해제할 수 있습니다.



키 사용법에 대해

Esc 키와 더불어 가장 많이 누르는 키입니다. 앞서 설명과 같이 Program Memory를 할 때 몇 번 누르냐에 따라 Program Memory의 남아있는 값이 달라지기 때문에 잘 숙지하고 계셔야 합니다.

Clear - 한 번 누르면 Select 된 상태가 풀립니다.

Clear **Clear** - 두 번 누르면 Selcet 된 Preset 상태가 취소됩니다. 예를 들면 메모리에 저장하고자 하는 값이 포지션 값 만인데 디머값이 같이 들어가 있다면 Clear를 두 번 누른 뒤 다시 디머값만 선택해 줄 수 있습니다. 주의할 점은 아직 무대 위의 그림은 사라지지 않았고 Active 값도 존재한다는 것입니다.

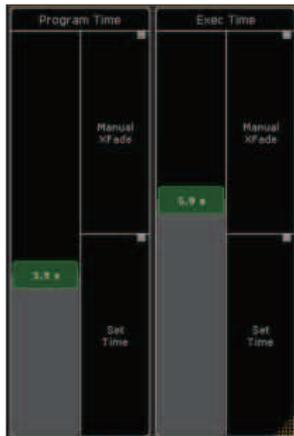
Clear **Clear** **Clear** - 세 번 누르면 콘솔 상의 수치값과 무대 위의 모든 값을 털어줍니다. 붉은색으로 된 상태도 회색(디폴트값) 상태로 돌아갑니다. Clear 세 번에도 이상한 값이 들어가 있다면 ESC도 눌러줍니다. 그래도 어떤 값이 남아있다면 Executor로 입력된 값일 것입니다.

Executor Memory

Executor는 실행하다, 실행하는(집행자) 등의 사전적 뜻을 가지고 있습니다.

즉, Program Memory 된 값을 실행하는 방법, 순서 우선순위 등을 설정하여 조명값들을 실행시킬 수 있습니다. 3개의 버튼과 1개의 페이더로 구성되어 있으며, Light 버전의 콘솔 기준으로 한 섹션당 15개씩 있습니다. 이 페이지를 넘겨서 진행할 수 있습니다. 또한 Executor는 독립된 버튼으로도 존재합니다. 개별버튼을 눌러서 진행할 수도 있습니다.

콘솔 Pool 창의 Sequence 탭에 연동되어 있습니다. Delete 키를 눌러 지울 수 있지만 완전히 지워진 것은 아니며, Sequence에 있는 값을 지워주어야 완전히 지워진 것입니다.



<수동으로 Program, Executor Memory Time을 지정할 수 있는 메뉴>

Onpc : Cmd section Pool 창

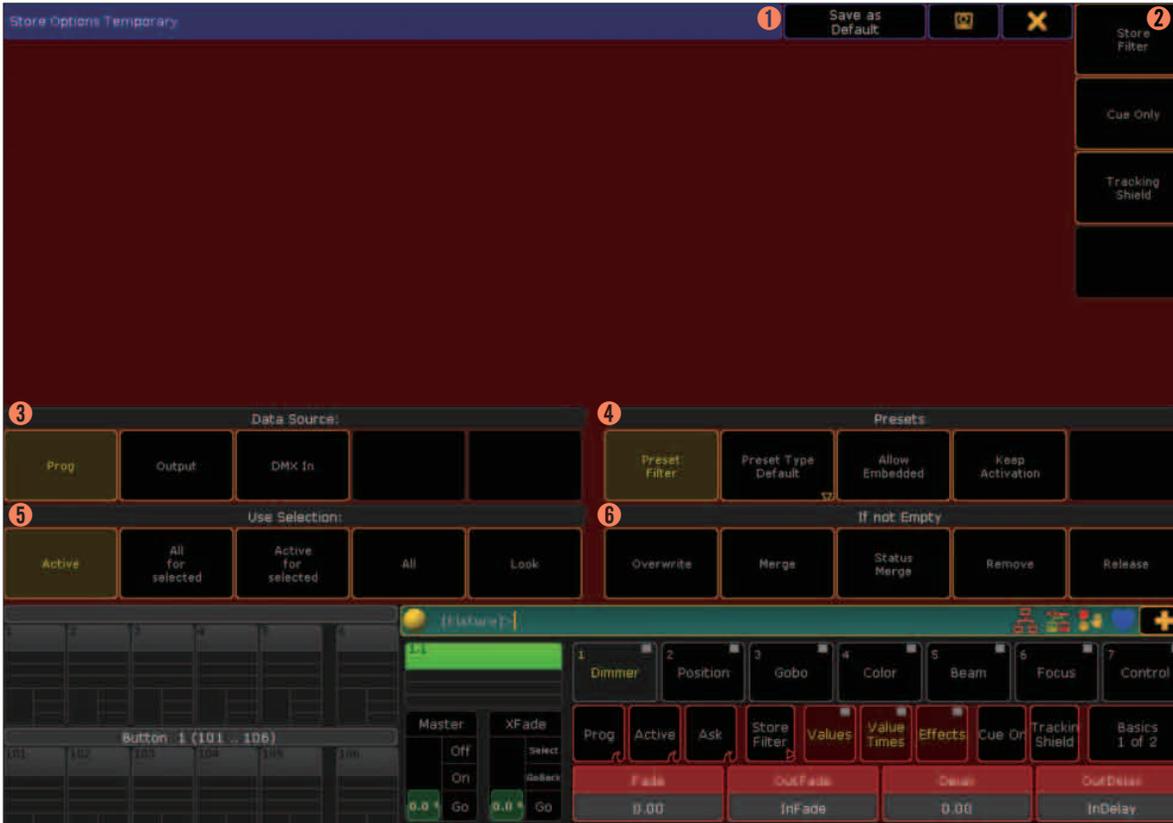
콘솔 : 2번 창 우측 하단

Set time 활성화 후 수동으로 time 조절 가능합니다.

이를 이용하여 메모리에 리듬감을 줄 수 있습니다.

Store

GRAND MA 2에서 가장 중요한 키 중의 하나입니다. 스토어 옵션값을 어떻게 정하느냐에 따라 Program, Executor로 들어가는 값이 달라질 수 있습니다. 이로 인해 의도와는 다른 메모리가 되기도 합니다. 그만큼 중요하기에 본격적인 메모리 과정에 앞서 Store option에 대해서 다루겠습니다.



<1번 섹션>

Save as Default : 원하는 값을 선택해준 후에 누르면 그 값이 디폴트값으로 저장이 됩니다. 자주 쓰는 방식이 생긴다면 해당 옵션들을 선택한 다음에 디폴트로 저장을 해주는 것이 좋습니다.

창 모양 아이콘 : 콘솔의 어느 창으로 보낼 것인지 지정할 수 있습니다. 콘솔 기준으로 보통 1번 창에 나오며, 크게 보고 싶다면 2,3번 창으로 옮겨주면 됩니다.

X 아이콘 : 해당 창을 닫습니다.

<2번 섹션>

Store Filter : 보통 모든 값이 다 들어가 있습니다. 스토어 필터를 선택하면 원하는 값(디머, 포지션, 포커스 등)을 걸러서 메모리할 수 있습니다.

Cue only : 해당 큐에만 스토어 값을 저장할 수 있습니다.

Tracking Shield : 트랙킹 모드를 보호해줄 수 있습니다.

<3번 섹션 Data Source>

저장되는 수치 값의 속성을 정할 수 있습니다.

Prog : 프로그래머가 만진 값을 저장합니다.

Output : Output 값이 저장됩니다. 즉, Dim값이 올라간 값으로 저장됩니다.

DMX In : 패치된 모든 값이 저장됩니다.

<4번 섹션 Presets>

Presets의 저장방법들을 정할 수 있습니다.

Presets Filter : 고유의 프리셋 값만 저장이 됩니다. Position pool 창에는 Position 값만 저장이 되는 것입니다. 즉, 이 기능을 풀고 저장을 하면 gobo Pool 창에 Dim이나 Position 값을 저장할 수도 있는 것입니다. 상황에 따라 편할 수도, 불편할 수도 있습니다.

Presets Type Default : Selective, Global, Universal 3가지의 방법을 정할 수 있습니다.

Selective : 선택된 조명기의 종류의 값을 저장합니다.

Global : 같은 종류의 조명기별로 저장합니다.

Universal : 조명기의 선택, 종류에 상관없이 모든 값을 저장합니다. 그러나 개별 조명기의 프로토콜 값이 다르기 때문에 보편적인 값, 디머값, 컬러값 등을 저장하는데 사용합니다.

Allow Embedded : 활성화 시켜 놓으면 개별 프리셋을 조합하여 All Presets에 저장할 경우 All preset을 수정할 경우 개별로 저장된 다른 값도 수정이 됩니다.

Keep Activation : 빈 창에 Presets을 저장할 경우 Selection이 풀리지 않습니다.

<5번 섹션 Use Selection>

선택 방법에 대해 설명합니다. 수치 값이 켜져 있더라도 어떻게 선택하여 저장하는 것에 따라 저장형태가 달라질 수 있으므로 매우 중요합니다. 흔히들 이 방법을 선택하는 성향에 따라 프로그래머의 메모리 방법, 스타일 등이 매우 달라질 수 있습니다.

Active : Active 된 모든 값을 저장합니다.

All for selected : 선택된 조명기들의 모든 Presets 값을 저장합니다. 예를 들어 1번 조명기를 Dim 값만 켜더라도 모든 Presets 속성이 저장됩니다. 즉, 만지지 않은 값은 0의 값으로 메모리가 됩니다. 만지지 않은 값도 메모리가 되기 때문에 이를 주의하여야 합니다.

Active for selected : Active 된 값 중 사용자가 선택한 값만을 저장합니다. 메모리의 마지막 단계에서 저장될 조명기를 선택해야 합니다. 손이 많이 가지만 가장 안정적이고 확실한 방법입니다.

All : 콘솔의 모니터 상에 존재하는 모든 수치 값이 메모리에 저장이 됩니다. 의도하지 않게 살짝 올린 값도 들어가기 때문에 주의해야 합니다. 또한 자주 사용할 경우 데이터 용량이 과도하게 커질 수 있습니다. 이를 주의하여 메모리 하여야 합니다.

Look : 눈으로 보이는 값이 저장이 됩니다. Look의 경우 Dim 값을 기준으로 1%라도 Dim이 올라가 있다면 해당 조명기의 모든 속성이 저장이 됩니다. 다른 값들을 건드렸더라도 Dim 값을 0으로 하고 저장한다면 해당 조명기의 다른 속성은 저장되지 않습니다. 이 경우 Dim 0 값만 저장이 됩니다.

<6번 섹션 If not Empty>

내가 저장하려는 값이 빈 곳이 아니라 무언가가 존재해 있는 곳에 저장할 때 정하는 옵션입니다. 켜 놓을 수도 있고 꺼 놓을 수도 있습니다. 메모리의 상황이 항상 같지 않기 때문에 꺼 놓고 하는 것이 좋습니다. 해당 상황이 왔을 때 판단하여 저장할 수 있습니다.

Overwrite : 새로 저장하는 값이 기존의 값을 지우고 덮어쓰는 방식입니다.

Merge : 기존의 값에 새로 저장하는 값을 병합시켜주는 방식입니다.

Status merge : 기존의 값에 새로운 값을 병합시켜주는데 상태(속성)를 같이 저장해주는 방식입니다. merge의 경우에는 트래킹 값이 그대로 따라가지만 Status merge는 트래킹 값이 빠지고 새로 저장하는 CUE가 기준이 됩니다. EFFECT 값이 제거가 안 되고 트래킹으로 따라올 때 흔히 많이 사용하는 방식입니다.

Remove : 해당 큐에서 제거하는 방식입니다. 따라오는 트래킹 값은 유지가 됩니다.

Release : 해당 큐에서 아예 빼는 방식입니다. 아예 수치상으로 없는 값으로 만들어 줍니다.

이상으로 Store option에 대하여 알아보았습니다. 스토어 옵션에 대해서는 잘 알고 있는 것이 중요합니다. 이 값을 잘못 지정하면 메모리 과정이 많이 꼬일 수 있습니다. 항상 숙지하고 계시는 것이 좋습니다.

1. 이제 이를 바탕으로 다음과 같이 스토어 옵션을 저장하겠습니다.

[예제 1] Store option 구성하기



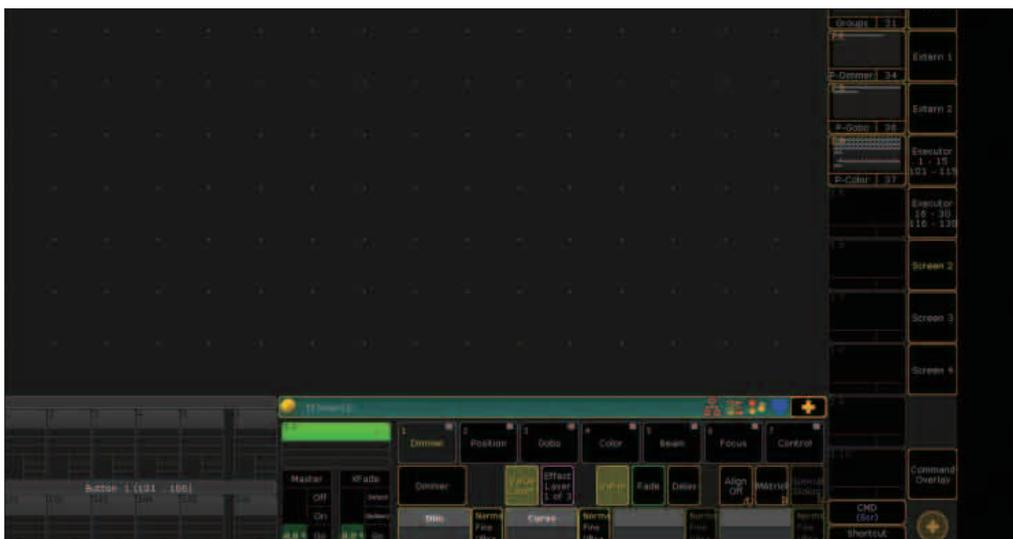
이와 같이 설정 후 Save as default를 눌러줍니다.

2. 창 구성하기

빈 창을 한 창에서 잘 볼 수 있도록 구성해줍니다.

[예제 1] 창 구성하기

1. 빈 창을 엽니다.(2번 창)



2. 빈 창에 다음과 같이 구성합니다.



예제에 활용할 수 있는 KEY

Tip

Move **A Click B** : A를 B로 이동시킵니다. 이를 이용하여 Color Preset을 보이기 쉽게 정렬해 줄 수 있도록 합니다.

Assign **Assign** **= Label** : 해당 프리셋의 이름을 바꿀 수 있습니다. 이를 이용하여 GROUP 이름을 바꿔줄 수 있습니다. 지난 시간 Create 애로 잡은 조명기들의 이름을 바꿔줄 수 있도록 합니다.

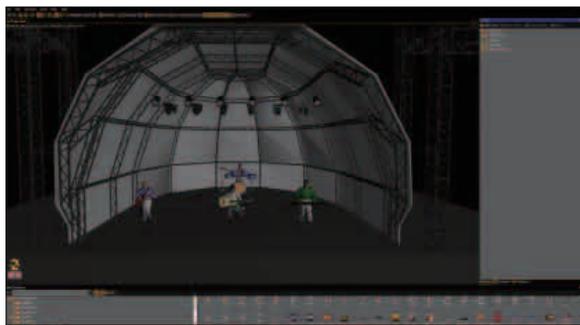
Assign **Assign** **Assign** **= Appearance** : 해당 프리셋의 외곽 테두리 색깔을 바꿀 수 있습니다. 이렇게 한 창에 프리셋들이 모여있을 때 구분하기 쉽게하기 위함입니다.

3. Preset 메모리 하기

이렇게 구성된 창을 바탕으로 큐 메모리를 해줄 수 있도록 합니다.

[예제 1] MA 3D에서 조명기 배치하기

지난 시간에 사전 배치한 조명기를 배치할 수 있도록 합니다.



3D 배치 후 확인 모습

콘솔의 Stage 창과 연동하면 더 쉽게 배치할 수 있습니다

Tip

밴드 기준으로

1. X축 값 : 센터 기준으로 -3m~+3m 안에서 일정하게 배치합니다.
2. Y축 값 : 맨 뒷줄부터 LED, BEAM, DIMMER 순서대로 배치합니다.
3. Z축 값 : 맨 바닥 기준으로 5m로 배치합니다.



콘솔 창과 연동 모습

Tip Set up을 누르면 Encoder 모양이 조명이거나 구조물을 옮길 수 있도록 바뀝니다. 이를 움직여 배치할 수 있도록 합니다. 작업 후에 꼭 Set up을 끄고 작업해야 원래대로 돌아갑니다.

[예제 2] 포지션별로 메모리 하기

1. 각각의 밴드 포지션으로 조명기를 이동시킵니다.
2. 이동 후에 Position Pool 창에 Store로 입력시켜 줍니다.
3. LED와 BEAM Position은 겹치기 때문에 Merge 옵션으로 병합시켜 줍니다.



각각의 조명기 포지션 확인

Tip MA 3D의 Edit - Follow 기능을 활용하면 3D 상에 보이는 빨간 점으로 조명기 포지션을 모아게 할 수 있습니다. 단, 조명기가 돌 수 있기 때문에 살살 움직여야 합니다.

[예제 3] Executor에 기본적인 CUE 넣기

Executor에 Store로 넣어줍니다.

[KEY 순서]

원하는 Program Memory로 만든 다음에

1. **store** → 2. 빈 Executor 클릭 → 3. **select** → 4. 메모리 한 Executor 클릭

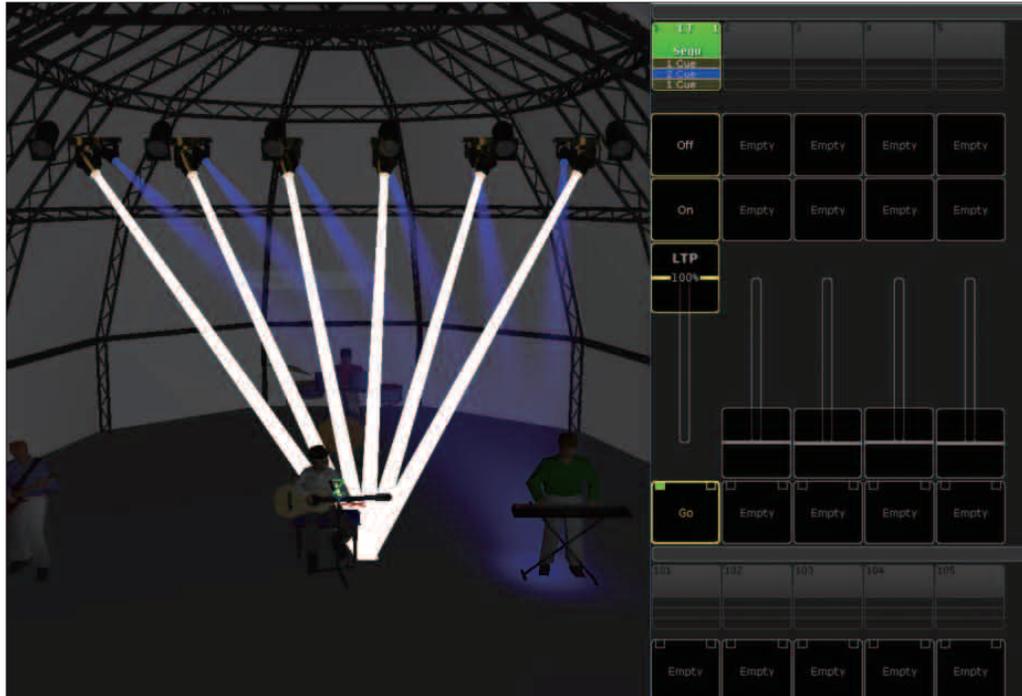
[예제 큐 순서]

CUE 1 - LED ZOOM 가장 작게 줄이면서 VOCAL POSITION 이동

CUE 2 - LED BLUE로 바꾸고, BEAM GUITAR POSITION 이동

CUE 3 - BEAM VOCAL로 이동하면서 RED Color Wheel로 바꾸어 주기

이렇게 포지션을 이동하면서 간단한 CUE 작업을 해봅니다.



큐 확인 모습

마치며

이번 편에서는 간단히 큐를 만들어 메모리해 보는 시간을 가졌습니다. 다음 편에서는 만든 큐들을 편집하고 큐 안에서의 타이밍 수정, MIB 등 큐를 전체적으로 편집하고, MIB를 활용하는 방법 등에 대해서 설명할 예정입니다. 아무쪼록 현장에서 작업하시는 데에 많은 도움이 되었으면 합니다. 문의 사항이나 궁금하신 사항이 있으면 kwb0323@naver.com으로 문의 주시면 소통할 수 있도록 하겠습니다. 감사합니다. 🙏