

**『송출용 비디오 서버 개발 공동연구』**

**수행 계획서**

**2015. 08.**

**(주)넷브로**

< 산출물 식별 >

|  |  |
| --- | --- |
| 프로젝트명 | 송출용 비디오 서버 |
| 발행년월일 | 2015년 08월 19일 |

< 산출물 검토 >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 소속회사 | 구분 | 성명 | 서명 | 일자 |
| (주)넷브로 | 작성자 | 조동현 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

< 문서 개정 이력 >

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 버전 | 변경일 | 변경사유 | 변경대상 페이지 | | 변경내용 | 승인 |
| 이전 | 이후 |
| V1.0.0 | 2015.08.19 | 최초 적용 | N/A | N/A | 변경내용 없음 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

< 목 차 >

[1. 개요 4](#_Toc428539657)

[1.1. 프로젝트 소개 4](#_Toc428539658)

[1.2. 시스템 구성 4](#_Toc428539659)

[2. 주요 개발 내용 6](#_Toc428539660)

[2.1. 비디오 서버 엔진 설계 6](#_Toc428539661)

[2.2. 출력 기능(Play out) 개발 11](#_Toc428539662)

[2.3. 입력 기능(Ingest) 개발 15](#_Toc428539663)

[2.4. 관리 S/W 개발 17](#_Toc428539664)

[2.5. 외부 제어 인터페이스 22](#_Toc428539665)

[2.6. 통합 실험 및 안정성 검증 24](#_Toc428539666)

[3. 추진 계획 일정 24](#_Toc428539669)

[4. 투입 인력 및 소요 예산 25](#_Toc428539670)

[4.1. 개요 26](#_Toc428539671)

[4.2. 공동 개발 연구비 상세내역 26](#_Toc428539672)

[4.3. 참여 인력 및 담당업무 27](#_Toc428539673)

# 개요

## 프로젝트 소개

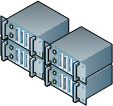
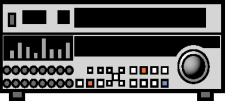
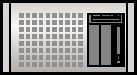
KBS 제작/송출 환경은 방송 시스템 변화에 따라 디지털 환경의 프로세스로 급격히 전환되어 왔다. 따라서 KBS 사내의 방송환경 변화에 유연하게 대응하고, 디지털 기반의 워크플로우를 효율적으로 개선하고, 송출 업무의 효율성을 높이기 위한 송출용 비디오 서버에 대한 개발이 요청되어 왔다.

본 사업은 KBS내의 송출 워크플로우 효율화 및 운영의 안정성을 제공하기 위해 방송용 파일 포맷에 대한 SDI 입출력이 가능한 송출용 비디오 서버 개발사업으로, 주조 작업자의 요구사항이 최대한 반영된 비디오 서버의 구조설계, 송출 서버의 기본 기능인 입력 기능, 출력 기능, 외부 제어 기능, 사용자에 최적화된 관리 소프트웨어 개발, 송출용 비디오 서버의 안정성 검증을 위한 통합 실험으로 진행되며, 향후 스튜디오에서 다수의 카메라 출력을 파일로 인제스트하기 위한 스튜디오용 비디오 서버와 제작편집실의 VCR을 대체하여 풀 파일 기반 워크플로우를 구현할 수 있도록 하는 제작용 비디오 서버까지 다양한 기능의 비디오 서버의 기반 기술을 확보하기 위한 공동 연구를 진행하는 것을 목표로 한다.

## 시스템 구성

**Playback**

* Configuration
* Play Control
* Monitoring



**Recording**

* Configuration
* Ingest Control
* Monitoring



Camera

VCR



**Video Server**

**Contents Management**

* Meta data
* Transfer Control



MAM & Storage

Monitor

비디오 서버는 비디오 입출력을 담당하는 엔진부가 있고 엔진을 제어하는 어플리케이션부로 구성되며, 어플리케이션 파트는 외부 연동을 위한 네트워크(UDP/IP)와 시리얼 포트(VDCP)를 지원한다.

또한 컨텐츠를 관리하고 외부 MAM 시스템에 전송하는 컨텐츠 관리 프로그램과 연동되어 동작한다.

* 녹화 및 재생용 Application : Playback/Recording 모듈로 구성되며, Playback/Recording Engine과 통신하여 컨텐츠 생성 및 재생을 할 수 있음. 녹화 및 재생용 Application은 Ethernet 및 RS422 제어 포트를 이용하여 외부 장비 연동이 가능하도록 구성되며, 컨텐츠 Management 모듈을 이용하여 컨텐츠 저장 및 재생을 할 수 있도록 구성
* 외부 시스템 연동 모듈(Ethernet/Serial Remote Protocol) : 비디오 서버에 대하여 외부 장비를 이용하여 제어할 수 있도록 원격 제어용 Native 및 표준 Protocol 지원
* 컨텐츠 관리 모듈 (Contents Management) : 비디오 서버에서 생성된 컨텐츠를 저장 또는 불러올 때 사용

# 주요 개발 내용

## 비디오 서버 엔진 설계

fig.1 Matrox Engine 동작 개요

1. 방송 신호 입출력 보드에 대한 SDK 분석
   1. X.MIO HW 구성

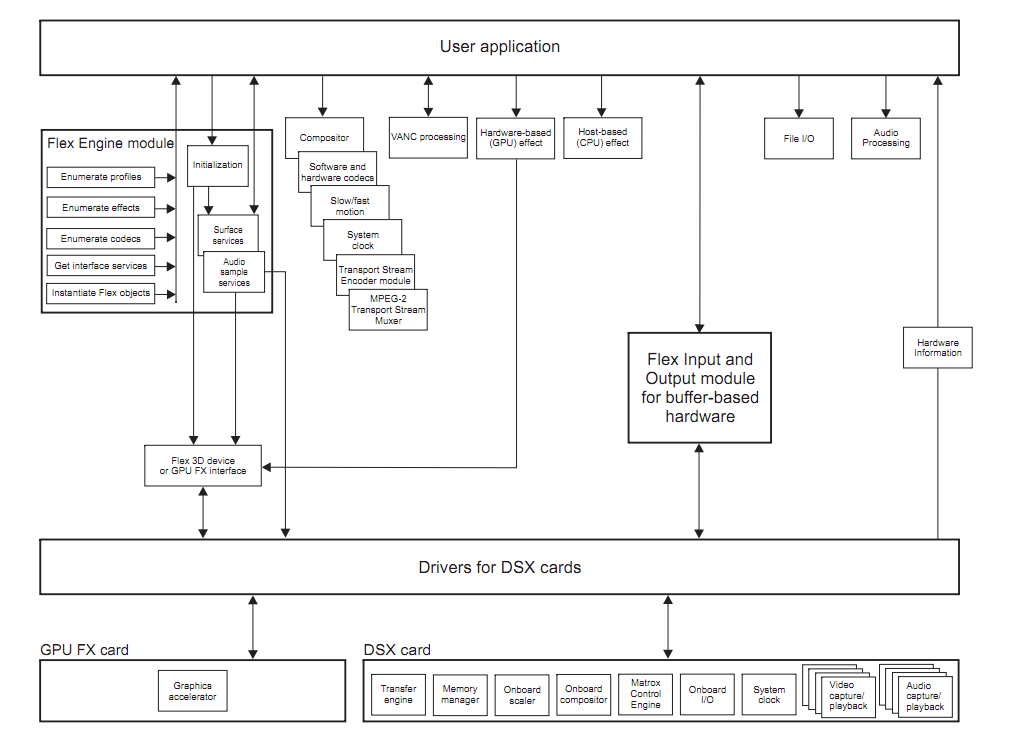


fig.1-1 Matrox Hardware 구성

* + 1. Matrox hardware는 물리적 HW와 상위에 드라이버 레벨, 그 위로 사용자가 직접 제어할 수 있는 application 레벨로 구분된다.
    2. fig.1-1에 보이는 바와 같이 Flex engine Module은 개발자가 실제로 H/W를 제어할 수 있는 전반적인 환경을 제공한다.
       1. 이는 H/W의 frame buffer, clock, composer 등 개발자가 사용할 수 있는 board내 기능의 interface를 제공한다.
       2. 각 기능별 설명은 추후에 기술한다.
    3. Flex Input and Output 모듈은 파일 입출력, IO입출력을 하기 위한 H/W에 접근할 수 있는 환경을 제공한다.
  1. X.MIO 하드웨어 제어
     1. IO Port
        1. X.MIO Board에는 비디오 입출력 및 reference signal 등을 받기 위한 다수의 IO port가 존재한다
        2. Matrox Input 포트는 실질적 입력을 담당하는 물리적 포트이며, Matrox Output 포트는 실질적 출력을 담당하는 물리적 포트이다.
        3. 각 포트의 resolution, frame rate, ref signal의 설정을 통해 원하는 출력을 수행한다
        4. Audio signal의 경우 SDI embedded 혹은 AES/EBU를 선택하여 입출력이 가능하다
     2. Frame Buffer
        1. fig.1의 예와 같이 입력된 신호는 보드 내의 IO 칩셋을 통해 frame buffer에 데이터가 저장된다.
        2. 이때 frame data는 matrox 하드웨어 영역 혹은 시스템 로컬 메모리에 저장 할 수 있으며 용도에 따라 이동하여 원하는 작업을 수행하게 된다
        3. Frame buffer를 사용하는 방식은 C. X.MIO 소프트웨어 제어 항목에서 설명한다.
     3. Mixer / Composer
        1. 이 데이터는 오디오/비디오 별 mixer와 composer를 이용해 가공할 수 있다
        2. 예를 들어 matrox board 내의 칩셋을 이용해 데이터를 가공하는 경우 데이터는 board 내의 메모리로 이동되어야 한다.
        3. 가공된 혹은 원본 그대로의 프레임 데이터는 matrox reader 혹은 writer를 통해 저장소에 저장하거나 읽어올 수 있다.
        4. 저장, 읽기의 경우는 3.의 항목과는 반대로 시스템 로컬 메모리에 존재하며 matrox board의 기능을 사용하기 위해서는 IO board 메모리로의 복사 또는 이동이 필요하다
        5. 이 과정은 C. X.MIO 소프트웨어 제어 항목에서 설명한다.

fig.2 Event 동작의 예

* + 1. 하드웨어 Feedback (fig.2)
       1. Signal의 유무
       2. Ref signal의 유무
       3. 입력된 H/W의 feedback은 event형태로 S/W 레벨로 전달된다.
       4. Event의 발생 시 엔진 코어는 어플리케이션 레벨에 이를 통보한다.
       5. 어플리케이션은 event의 경중에 따라 사용자에게 직접 알리거나 로그를 남긴다.
    2. Matrox clock
       1. Nano초 단위의 시간을 이용하여 드랍 프레임 계산 및 송출 시 프레임당 정확한 time stamp를 설정한다.
  1. X.MIO 소프트웨어 제어
     1. Matrox Surface , Samples
        1. Matrox의 전반적인 녹화와 재생과정의 이해를 위해서는 surface와 Sample의 이해가 필요하다.
        2. Surface는 영상, Samples는 음성의 데이터로 이해하면 됨
        3. 2.1.1.B.ii에 기술한 frame buffer 데이터에 접근하기 위해서는 matrox API의 surface를 이용하여야 한다.
        4. 큰 의미로 프레임 이미지를 의미
        5. ARGB, YUVA간 변환이 가능하며 이를 이용하여 이미지의 변환, 합성에 사용된다.
        6. 논리적인 matrox SDK상의 이미지 버퍼로 로컬 메모리와 Board IO 메모리의 frame buffer를 제어할 수 있다.
        7. Frame date는 surface 형태로 접근할 수 있으며 surface의 위치, 타입 등을 지정해 생성 혹은 생성되어있는 데이터의 접근이 가능하다
        8. ex) new surface(Location, ColorFormat...)
           1. Location은 surface의 생성 위치, ColorFormat은 surface의 색상 영역을 지정할 수 있다.
           2. Surface의 생성 위치는 크게 system local memory와 board IO memory가 있다.
     2. Matrox Reader
        1. 시스템 저장소에서 파일을 읽어올 때 사용
        2. 파일의 Open시에 코덱 판별을 자동으로 함
        3. ex) GetBuffer ( VideoData, VideoDataSize, AudioData, AudioDataSize, SamplesPerBuffer, VBIData, VBIDataSize...)
           1. VideoData는 progress와 interlaced의 경우에 대응하기 위해 array의 형태로 구성
           2. AudioData 역시 VideoData에 대응하기 위한 array로 구성되어짐
           3. 기타 VBI Data등도 이 과정에서 읽어지게 됨
        4. 읽어진 데이터는 frame data 형태로 로컬 메모리에 존재한다.
           1. 저장장치의 파일을 읽어올 때 안정적인 reading을 위해 버퍼 사이즈를 조정
           2. (ex) SetPipelineSize(size)

이때 size는 개발자가 원하는 안정적으로 동작하기 위한 수치를 입력

사용자 요구에 따라 설정 값을 대입할 수도 있음

* + - 1. 재생방식(normal play, FF, REW, scrub, various play)에 따른 reader의 조절로 원활한 재생 지원
         1. Play() : reader를 플레이 상태로 설정
         2. Stop() : reader를 정지
         3. Seek(frame) : frame에 지정한 프레임으로 이동
         4. Shuttle(dValue) : 원하는 속도로 재생을 제어할 수 있음
         5. 상기 네가지 함수의 조합으로 재생 과정이 이루어진다.
         6. ex) 플레이의 경우 :

m\_pJFlexreader->shuttle(1.0)

m\_pJFlexreader->Play()

* + 1. Matrox Writer
       1. 1.A.에 기술한대로 입력된 frame data는 matrox IO board 내의 메모리에 존재하며 이 데이터의 저장을 위해서는 로컬 메모리로 복사 도는 이동되어야 한다
       2. writer는 로컬로 복사된 frame data를 저장소에 저장하는 역할을 수행한다
       3. 원활한 녹화 작업을 위해 파라미터를 세팅
       4. 저장 format, bitrate, frame rate 등의 설정이 필요
          1. 저장 미디어의 세팅은 해당 파일의 Open시에 이루어진다
          2. ex) OpenFile(FilePath, CaptureType, ResolutionInformation, VideoParameters, AudioParameters...)
          3. CaptureType으로 저장되어질 파일의 타입, 다시 말해 포맷을 결정한다(넓은 의미로 wrapper를 지정함)
          4. VideoParameters는 essence를 설정함. 실질적인 비트레이트 등의 설정은 여기에서 이루어짐
          5. AudioParameters는 VideoParameters의 개념과 동일
       5. 실제 저장 과정은 다음과 같다
          1. GetBuffer ( VideoData, VideoDataSize, AudioData, AudioDataSize, SamplesPerBuffer, VBIData, VBIDataSize...)

Reader의 사용과 동일하게 GetBuffer를 사용하여 얻어 옴

* + - * 1. CopySurface(SourceSurface, TargetSurface)

SourceSurface는 위의 GetBuffer에서 얻어온 write를 위한 surface이고 TargetSurface는 저장 될 surface이다.

surface와 samples에 저장하고자 하는 surface와 samples를 복사하여 저장 과정이 이루어진다

* + 1. On board compositor의 사용
       1. Mix / Compose의 수행에 사용
          1. GetCompositorInterface(Compositor instance)
          2. instance->UpstreamCompose( JDstSurface, iLayerCount, apISurfaces,...)

UpstreamCompose 함수를 통해 믹싱과 컴포징 과정이 이루어지며 JDstSurfacesms 출력 결과물이 저장되는 surface, apISurfaces는 다수의 surface를 한꺼번에 처리를 가능하게 한다

1. Engine Core 구성

각 채널 별 독립적인(타 채널에 구애 받지 않는) 역할 수행을 위해 독립적인 프로세스로 동작

* + 1. 엔진은 채널 프로세스의 독립적 동작을 염두 하여 설계
    2. 독립적 구조하의 동작에서 채널 간 간섭이 없어야 함

1. Matrox Input, Output
   1. 영상, 음성의 입출력을 담당하는 인터페이스
   2. 해당 인터페이스의 접근 방법은 다음과 같다.
      1. GetInputOutputIndependentInterface(InputOutputIndependent )
         1. IO 인터페이스의 접근을 위한 인스턴스를 얻는다
      2. m\_pJInputOutputIndependent->GetVideoPlaybackInterface(index, PlaybackControl )
         1. Playback 인터페이스를 얻어온다.
         2. index : 얻어오고자 하는 채널의 번호
         3. PlaybackControl : 제어하고자 하는 output의 인스턴스가 할당된다.
      3. m\_pJInputOutputIndependent-> GetVideoCaptureInterface (index, CaptureControl)
         1. Playback 인터페이스를 얻어온다.
         2. index : 얻어오고자 하는 채널의 번호
         3. CaptureControl: 제어하고자 하는 input의 인스턴스가 할당된다
      4. 입력된 frame data는 board 상에 존재(IO surface, IO samples)
      5. 출력하고자 하는 frame data도 board에 존재해야 한다.
      6. Play의 경우
         1. PlaybackControl->PlayBuffer(SrcSurface)
         2. SrcSurface는 출력될 surface
      7. Record의 경우
         1. m\_ CaptureControl->CaptureBuffer(CaptureSurface)
         2. CaptureSurface는 입력된 surface

## 출력 기능(Play out) 개발

fig.4 Play out 과정 개요

1. 미디어 흐름 및 개요
   1. Flex Reader를 이용해 저장장치에서 미디어 데이터를 읽어 들임
      1. 상기한 Buffer및 여러 파라미터의 조절로 안정적인 플레이 지원
      2. reader의 제어를 통해 Play, Stop, Pause, FF, REW등을 설정
         1. 상세 설명은 2.1.1.C.ii.5 참조
      3. 이 과정은 이론적으로 명령이 들어온 다음 프레임의 시점부터 적용됨
   2. 읽어진 데이터는 matrox output을 통해 baseband signal로 변환되어 출력
      1. reader에서 읽어진 frame data는 2.1.1.C.i항목의 surface형태로 존재한다. (이는 video data의 경우이고 audio data는 samples의 형태로 존재)
      2. 이 surface는 로컬 메모리(host)상에 존재한다.
      3. host상의 surface는 직접 송출이 불가능하므로 board 상의 메모리(IO)에 복사 또는 이동되어야 한다
         1. ex) CopySurface(HostSurface, IOSurface)
      4. 각각의 audio, video데이터는 최종 출력할 때 까지 각각 분리되어 처리한다.
      5. IO 상의 데이터는 board의 처리 칩셋을 이용할 수 있으며 board H/W가 지원하는 범위 내에서 mix, compose의 과정을 거친다.
         1. 2.1.1.c.iv 항목참조
      6. 처리과정이 끝난 데이터는 matrox output을 통하여 베이스밴드 형태의 시그널로 출력되며 원하는 경우 AES/EBU를 통한 audio 출력이 가능하다.
      7. 분리된 데이터의 sync 및 정확한 프레임 데이터의 재생을 위해 time stamp를 이용한다.
         1. ex) surface->SetTimestampInfo(stamp)
            1. 이때 stamp는 나노초 단위의 프레임이 갖는 시간이다.
   3. 미디어 제어
      1. Reader의 제어
         1. 1.A에 기술한 대로 재생에 관한 모든 제어는 reader의 파라미터를 조절하여 이루어진다.
         2. 각 채널 별 재생 클립에 대한 Play, Stop, Pause, FF, REW 제어
         3. 재생 클립에 대한 +1 Frame, -1 Frame 제어
         4. 재생 클립에 대한 Scrubbing 검색
         5. 2.1.1.c.ii.5 항목 참조
   4. 재생과정

fig.5 재생과정(SW) 흐름도

* + 1. fig.5는 한 프레임을 재생하는 과정을 간략히 표현한 것이다
    2. 해당 프레임이 송출되는 타이밍은 처리 과정의 최초에 matrox clock을 이용해 얻어와 할당되어지며 이때 드랍 프레임의 검사가 이루어진다
    3. 미디어의 데이터를 얻어올 때 플레이 상태에 따라 어떤 프레임을 가져올지를 결정한다.
    4. Super out등의 기능이 필요한 경우 compose/mixing과정에서 수행된다.
       1. 2.1.1.c.iv 항목 참조
    5. 처리가 완료된 데이터는 baseband signal 혹은 AES/EBU의 형태로 출력된다.
  1. 재생상태 확인
     1. 재생상태 표시 : 대기(재생 종료), 재생, 중지, 배속 재생 등
     2. 출력신호 상태 표시 : 출력영상을 화면으로 표시, 출력음성 각 채널 별 레벨미터 표시
     3. 재생 중인 파일의 정보 표시 : 코덱, 해상도, 길이, 반복 재생 상태 등 표시
     4. 정상적인 플레이가 불가능한 경우 그에 대한 이유, 간단한 해결 방안 등 제시

1. 플레이리스트 구성
   1. 플레이 리스트 재생, 연속 재생, 구간 재생 등 확장 재생 제어 기능
      1. Matrox reader에 playlist를 구성
         1. Reader->GetPlayList(PlayList)
            1. Reader에서 playlist의 인스턴스를 얻어와 구성한다
      2. reader는 구성되어진 playlist를 기반으로 동작하며 reader가 동작하기 위해서는 반드시 플레이리스트가 존재해야 한다
   2. 파일 별 개별 송출 혹은 단위 별 송출 기능을 지원
      1. Matrox의 playlist는 위 기능을 기본적으로 지원하지 않기 때문에 어플리케이션 레벨에서 기능을 구성해 주어야 함.
         1. 총 구성된 플레이리스트 내의 어느 프레임이 어느 항목인지 어플리케이션에 저장하고 항목별 인덱스를 프레임에 적용하여 관리
   3. 리스트에 파일을 추가 시 mark in, mark out의 설정 가능
      1. 파일을 사용자가 원하는 혹은 APC에서 명령한 일부 구간만 리스트에 추가할 수 있는 기능
         1. Matrox playlist에 appendmedia 함수를 이용하여 해당 미디어의 원하는 구간만을 추가 할 수 있음
            1. ex) AppendMedia(FileName, MarkIn, MarkOut...)

FileName : 파일의 위치

MarkIn : 시작 프레임

MarkOut : 종료 프레임

* 1. Dynamic preview : 사용자가 IO의 할당 없이 파일 확인 및 in, out을 설정하여 리스트에 추가할 수 있는 기능
     1. fig.5의 재생 과정에서 IO output 기능을 제거한 software playback 컴포넌트를 이용하여 송출 중에도 in, out 편집, 확인을 가능하게 하여 운용상의 편의성을 도모하고 실수를 방지
  2. Dynamic playlist 지원 (재생중인 list에 항목 추가 삭제 가능)
     1. 송출 중에도 현 시점 이후의 항목에 대하여 추가, 삭제 가능

1. 외부 기준 신호 입력 및 동기화 기능 개발
   1. Ref signal을 통한 타 시스템과의 동기화
   2. Ref signal의 유무 체크 및 상태를 feed back
      1. fig.2의 이벤트 처리 과정 참조
2. Super out
   1. 사용자가 시스템 운영에 유용한 정보를 출력화면에 overlay하여 보여줌
      1. fig.5의 compose, mixing기능을 이용하여 진행상태 및 정보를 송출 중인 영상에 표시
      2. 이 기능은 할당된 원본 영상 출력 이외에 IO output port를 하나 더 필요로 함
   2. 서버의 재생 상태, 재생되어지는 클립의 코덱 등
   3. 필요한 정보는 사용자의 요구에 따라 추가, 삭제 될 수 있음
3. Multi-codec 지원
   1. Matrox가 지원하는 방송용 재생 코덱
   2. MXF(D10[IMX], XDCAM HD 35, XDCAM HD 50, Generic MPEG2 …)
   3. MOV(DVCPRO, DVCPRO50, DVCPRO HD, ProRes, …)
   4. AVI File(Generic MPEG2), WAVE File(48000Hz, 16 or 32bit)
4. Dolby-E pass-through
   1. Dolby Audio 처리를 위한 Audio 5.1ch pass-through 기능 구현

fig.5-1 Dolby-E pass-through 개요

* + 1. matrox input에 입력된 surface와 samples를 가공과정 없이 matrox output으로 bypass 기능 추가
       1. 2.1.1.c.v 항목에 설명된 matrox input, output을 사용
       2. Input을 통해 입력된 samples는 IO memory상에 존재
       3. 이 samples를 output으로 직접 출력 (2.1.1.c.v 참조)

1. 설정
   1. Resolution Setting : 입출력 해상도 크기를 설정
   2. Sync Setting : 동기화 신호를 입력 받아 처리할 수 있도록 설정
   3. 그 외 시스템에 필요한 설정을 추가

## 입력 기능(Ingest) 개발

fig.6 Ingest 개요

1. SDI Signal의 녹화 과정

fig6-1 녹화과정(SW) 흐름도

* 1. Input signal은 matrox input을 거쳐 io board내의 frame buffer에 저장됨
     1. 이때 저장된 data는 matrox surface의 형태로 저장
     2. 저장된 surface는 IO 상에 존재하므로 저장 작업을 위해서는 host로 복사 또는 이동이 필요함
     3. 복사/이동된 surface는 matrox writer를 통해 저장소에 저장
     4. 이후 저장된 surface의 baseband 출력은 super out 출력을 위해서 surface의 가공이 필요
        1. mix, compose작업을 위해 IO 상의 surface 이용
        2. 타임코드, 저장소의 남은 용량 등의 text 데이터를 이미지 데이터로 변환 한 후 on-board composer를 이용하여 surface에 overlay하여 baseband로 출력
  2. 이 과정에서 입력 단이 외부 타임코드를 받을 수 있으면 이를 저장할 수 있음
     1. RP188 형태로 입역되어지는 타임코드는 surface 형태로 변환되며 matrox writer를 통해 저장된다

1. 지원 포맷
   1. Matrox가 지원하는 방송용 멀티코덱 지원
      1. MXF(D10[IMX], XDCAM HD 35, XDCAM HD 50, Generic MPEG2 …)
      2. MOV(DVCPRO, DVCPRO50, DVCPRO HD, ProRes, …)
      3. AVI File(Generic MPEG2), WAVE File(48000Hz, 16 or 32bit)
   2. Matrox Writer Setting
      1. Writer는 파일의 open시 혹은 추가적으로 옵션을 설정할 수 있음.
         1. 이 단계에서 format, framerate, bitrate 등을 설정할 수 있음
2. 녹화의 제어
   1. 저장할 파일 설정
      1. 파일명, 2.B.항목의 옵션 설정
   2. 수동 녹화
      1. 사용자가 원하는 시점에 녹화 시작, 중단, 종료
      2. 일반적인 방법으로 사용자가 로컬 UI를 보고 이용하여 진행
   3. 자동 녹화
      1. 사용자가 원하는 타임코드 혹은 시간을 미리 설정 한 후 시스템이 자동으로 녹화를 진행
      2. 어플리케이션 레벨에서 동작하며 자세한 사항은 2.4.관리 소프트웨어 항목에서 설명
3. 녹화 설정
   1. 입력 영상의 해상도, 주사율, 입력 음성의 채널 수 설정
   2. 저장될 파일의 형식을 설정 : 비압축, 입력장치 또는 인코더가 지원하는 코덱 등
   3. 이때 저장된 파라미터는 matrox writer의 설정에 사용됨
      1. 세부사항은 2.B.1에 기술한 바와 같음
4. 상태 확인
   1. 엔진이 현재 수행하고 있는 상태를 feed back
      1. fig.2. 참조
   2. 엔진 상태를 모니터링 가능하게 함
   3. 입력신호 및 ref signal의 유무 feed back
   4. live window(영상), vu meter(음성)의 시각적 모니터링 지원

## 관리 S/W 개발

1. Local UI



Control Button

Audio Level Meter

Live Window

fig7. Local UI의 예

* 1. Live Window 및 Audio Level Meter
     1. 입/출력 영상의 Local mornitoring을 위한 UI 상의 영상 display
        1. 입력 surface를 로컬 시스템에서 사용 가능한 형태로 변환하여 모니터(VGA)에 표시
     2. 입/출력 영상의 Audio를 그래프로 시각화
        1. 입력되어진 audio sample을 byte buffer형태로 변환한 뒤 그래프 형태로 도식화
  2. 미디어 제어
     1. Playout
        1. UI상의 버튼 및 외부 컨트롤을 이용하여 Play, Stop, FF, REW, Jog, Shuttle 등의 기능을 수행
           1. UI와 외부 컨트롤 명령은 엔진에 전달하기 전 해당 프레임 이후 프레임에 영향을 미침
        2. 재생중인 아이템의 metadata 표시
           1. Matrox media info 혹은 자체 개발한 media component를 이용하여 사용자가 서버 운용에 유용한 정보를 제공

Format, bitrate, frame rate, file size 등

* + - 1. Play list

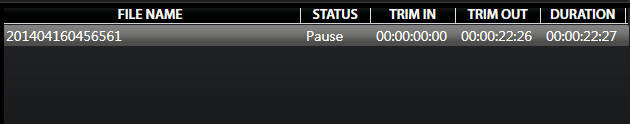


fig7-1. Playlist UI의 예

* + - * 1. 미디어의 en-cue, de-cue 플레이 리스트 내 추가 제거 등의 작업을 수행
        2. Drag & drop 지원

파일 추가 시 원하는 파일을 open하는 방식이 아닌 UI 내에서 drag & drop하는 방식을 지원하여 직관성과 편의성 제공

* + - * 1. Dynamic preview 기능을 통한 mark in, out의 지정 후 미디어 추가

2.2의 2.D항목 참조

송출 중에도 미디어의 preview를 통한 mark in, mark out을 지정 가능

미디어의 리뷰 후 작업을 수행하게 되어 작업 실수의 가능성을 줄이고 편의성을 높임

* + - * 1. 구성된 리스트에 대한 저장, 복사 등의 기능 지원

Preset형태의 스케줄 관리

불필요한 반복작업배제

시스템 장애 시 빠른 대처

시스템 장애가 생긴 경우 비어있는 채널 혹은 작업이 없는 장비로의 바른 전환 가능

* + - 1. Schedule list
         1. 지정된 시간에 원하는 미디어가 플레이 되는 기능 지원
         2. 구성된 리스트에 대한 저장, 복사 등의 기능 지원

불필요한 반복작업 배제

시스템 장애 시 빠른 대처

시스템 장애가 생긴 경우 비어있는 채널 혹은 작업이 없는 장비로의 빠른 전환 가능

* + 1. Ingest
       1. VCR
          1. VCR을 통한 tape녹화 시 Sony 9pin protocol을 통한 제어 가능

RS-422 Serial Communication

Protocol 연동을 통한 VCR-VCR 녹화와 같은 환경 제공

사용자가 VCR과 서버를 번갈아 가며 작동하는 불편함 감소

* + - * 1. Play, Stop, FF, REW등 기본적인 제어 외에 tape의 유무 등 상태확인 가능

Protocol packet의 status enum값을 이용하여 로컬상에서 상태를 확인하고 ingest작업을 진행

* + - * 1. 녹화 및 녹화된 미디어의 Preview 기능 지원

Baseband signal 출력을 통해 방송용 모니터로 프리뷰

로컬의 live window를 이용

Live window를 이용하는 경우 원본보다 많이 작아진 영상을 보게 되므로 방송용 모니터를 통한 리뷰를 추천

* + - * 1. Batch Ingest

한 테이프에서 하나 혹은 다수의 구간을 설정하여 녹화를 진행

이때 Sony 9pin protocol을 이용한 VCR연동

자동녹화 항목이 이에 해당

* + - * 1. Schedule Ingest

녹화 시작과 종료시점을 미리 정하여 해당 시간에 녹화하는 기능

자동녹화 항목이 이에 해당

* 1. 미디어 관리
     1. 로컬 미디어 관리
        1. 파일의 복사, 이동, 삭제의 기능을 지원
           1. 인제스트 시 작업 디렉터리의 관리
           2. Play out의 경우 파일 브라우저 기능 지원
        2. 일원화된 파일의 관리로 미디어의 유실 가능성 배제
           1. 파일 작업에 대한 이력을 log로 저장
        3. 사용자의 편의성 향상
        4. 파일 브라우저 이용하는 경우보다 직관적이고 빠른 작업 환경 지원
     2. 원격 미디어 관리
        1. 원격지에서 FTP 서버를 통해 미디어의 Upload, Download를 지원
        2. 사용자 권한 부여
           1. 사용자 별 미디어 관리로 부주의에 의한 파일 유실 방지 및 미디어 보호
  2. 상태 모니터링
     1. 엔진의 상태 체크 및 feedback을 기반으로 시스템의 상태를 분석하여 사용자에게 운용에 필요한 정보 제공
        1. fig.2. 참조
        2. Ref signal의 유무 표시
        3. Drop frame 등 운영 중 서버의 이상 유무 표시
           1. 2.2항목의 1.D 참조
        4. 스토리지 연결 상태 및 남은 용량 표시
           1. 주기적인 저장소의 용량 체크 혹은 매 작업 종료 시를 선택하여 남은 용량, 인제스트 시 대략적인 남은 녹화시간 등을 표시
        5. Log 파일 생성



fig.7-2 Log 파일의 예

* + - * 1. 로그는 종류에 따라 정상, 경고, 오류의 타입으로 구분되어 저장됨
      1. 기타 사용자 요구에 따른 정보 표시

1. 설정
   1. Local 환경설정
      1. 일반적인 어플리케이션의 로컬 UI를 통한 환경설정
         1. Playout
            1. resolution, frame rate
            2. 외부 제어를 받기 위한 serial 통신이나 socket통신
            3. 그외 요구에 따른 설정 항목 추가
         2. Ingest
            1. 녹화 포맷, 해상도, 프레임 등
            2. 외부 제어를 받기 위한 serial 통신의 설정, socket 통신을 위한 IP 설정 등
            3. 녹화 sync를 위한 설정 등
   2. Web Service 환경설정

fig8. Web Service 환경설정 개요

* + 1. 원격지에서 web service를 통한 환경설정
       1. Web browser를 통한 원격 환경설정
       2. Web service를 통해 입력된 환경설정 데이터는 XML형식으로 저장
       3. 서버 어플리케이션으로 string형태 혹은 파일 형태로 전달
       4. Configuration module은 설정을 적용할 타이밍을 사용자에게 feed back하고 엔진에 설정을 적용한다
    2. 사용자가 local UI를 일일이 다니며 접근해야 하는 불편을 해소
    3. 모니터링을 통시에 지원하여 보다 나은 관리 환경 제공

## 외부 제어 인터페이스

1. RS-422
   1. VDCP
      1. 개요

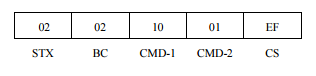


fig.9. VDCP Protocol의 구조

* + - 1. 기본적으로 command, bit count, data, checksum의 구조로 이루어져 있다.
      2. command는 byte data로 되어있으며 표준 VDCP Protocol에 정의된 규약을 따름

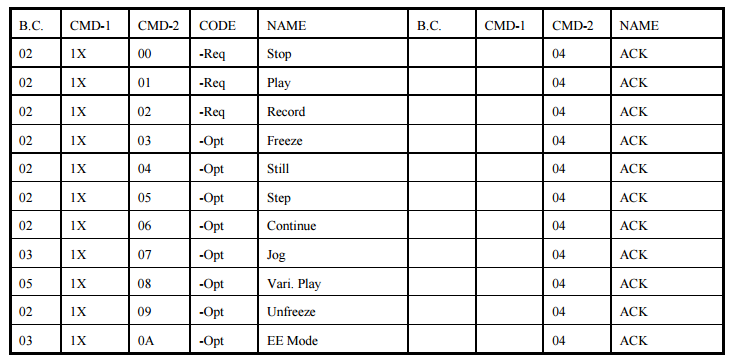


fig.10 VDCP 명령어의 예

* + 1. Server 제어 프로토콜의 지원으로 타 장비와 연동의 효율성 확보
    2. 기존 VDCP의 정의에 할당 되어있지 않은 명령은 확장하여 지원
  1. SONY 9 Pin
     1. Sony에서 제안한 RS-422을 통한 제어방식으로 다음의 경우에 사용한다
        1. VCR을 제어
           1. Ingest 동작 시 VCR을 표준 프로토콜을 통해 제어한다
           2. 패킷과 명령어의 구조는 VDCP외 유사함.
        2. Editor의 명령을 받아 동작하는 경우
           1. 1의 경우와 반대로 VCR을 대체한 시스템으로 동작해야 하는 경우
        3. 기타 VCR처럼 동작을 해야 하는 경우
           1. VCR과 유사한 패턴의 데이터 통신을 반드시 필요로 하는 장비의 경우 반드시 sony 9pin 제어를 이용해야 함

1. LAN(Socket) 통신
   1. 신규 장비 연동 시 속도, 거리와 효율 증가를 위한 TCP/IP 기반의 자체 프로토콜 지원
   2. API 형식으로 제공하여 개발 편의성 향상
      1. 기호 혹은 바이트 형태의 데이터가 아닌 문자열 자체를 명령으로 사용하므로 직관적인 개발 및 디버깅 환경 제공
         1. ex) play|F:\MXFFolder\TestClip001.mxf|00:05:00:00|00:10:00:00
            1. 해석 : F:\MXFFolder\TestClip001.mxf파일을 타임코드 00:05:00:00 부터 00:10:00:00까지 재생
            2. 위에 언급한 VDCP나 Sony 9pin처럼 명령에 제약이 없으므로 직관적이고 큰 사이즈의 데이터 통신이 가능함
            3. 타 장비와의 연동 개발 시 명령어 자체만으로 어떤 명령이 입력되었는지 판단 가능

## 통합 실험 및 안정성 검증

|  |  |
| --- | --- |
| **구분** | **개발 내용** |
| 입출력 기능 및 관리 기능 통합 실험 | * 각 기능별 개발자 단위의 단위 테스트 진행 * 사용자 중심의 전문 테스트 요원의 관리 소프트웨어 통합 테스트진행 * 각 단계별 테스트 시험 성적서 작성 및 오류 분석 |
| 비디오 서버 안정성 검증 | * 하드웨어 안정성 검증을 위한 Aging Test * 전원, Fan등 주요 이중화 부품에 대한 Hot-plug, Hot-swapping 기능 테스트 * 입/출력 컨텐츠의 Non-drop Frame 보장을 위한 안정성 Aging Test (최소 4주간 실시) * 소프트웨어 안정성 검증을 위한 전문 테스트 요원의 어플리케이션 테스트 |

# 추진 계획 일정

* 개발 기간 (개발 5개월, 안정성 검증 및 통합 실험 3개월)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **연구 내용 및 항목** | **추 진 일 정** | | | | | | | | **역할분담** | |
| **M1** | **M2** | **M3** | **M4** | **M5** | **M6** | **M7** | **M8** | **KBS** | **업체** |
| **I. 송출용 비디오 서버 개발** |  |  |  |  |  |  |  |  |  |  |
| **가. 기능 설계** |  |  |  |  |  |  |  |  |  |  |
| - 입출력 보드 SDK 분석 및 실험 |  |  |  |  |  |  |  |  | ◎ | ◎ |
| - 기능별 모듈화 설계 |  |  |  |  |  |  |  |  | ◎ | ◎ |
| **나. 출력 기능 개발** |  |  |  |  |  |  |  |  |  |  |
| - SDI 출력 제어 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - 확장 재생 제어 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - 외부 기준 신호 입력 및 동기화 개발 |  |  |  |  |  |  |  |  | ● | ○ |
| - 수퍼 아웃 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - MPEG2, Prores, MXF, MOV 포맷 출력 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - Dobly-E pass through 기능 개발 |  |  |  |  |  |  |  |  |  |  |
| **다. 입력 기능 개발** |  |  |  |  |  |  |  |  |  |  |
| - SDI 입력 제어 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - MPEG2, Prores, MXF, MOV 포맷 저장 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| **라. 관리 S/W 개발** |  |  |  |  |  |  |  |  |  |  |
| - 제어 및 모니터링 UI 개발 |  |  |  |  |  |  |  |  | ◎ | ◎ |
| - 다중 채널 입출력 관리 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - 내부 컨텐츠 관리 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - 외부 파일 전송 기능 개발 |  |  |  |  |  |  |  |  | ● | ○ |
| - 예외 상황 처리 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - 상태 로그 저장 및 내/외부 알림 기능 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| **마. 외부 제어 인터페이스 개발** |  |  |  |  |  |  |  |  |  |  |
| - RS-422 인터페이스 제어 프로토콜 개발 |  |  |  |  |  |  |  |  | ○ | ● |
| - TCP/IP 외부 제어 프로토콜 및 실험 S/W 개발 |  |  |  |  |  |  |  |  | ● | ○ |
| **바. 통합 실험 및 안정성 검증** |  |  |  |  |  |  |  |  |  |  |
| - 입출력 기능 및 관리 기능 통합 실험 |  |  |  |  |  |  |  |  | ◎ | ◎ |
| - 비디오 서버 안정성 검증 |  |  |  |  |  |  |  |  | ◎ | ◎ |
| **II. 착수/중간/최종 발표, 보고서 작성** |  |  |  |  |  |  |  |  |  |  |

# 투입 인력 및 소요 예산

## 개요

* + - 1. 전체 연구 개발비의 구성

: 직접인건비 및 제경비

* + - 1. 연구개발비용 분담

: 공동연구개발비 총액은 양분하는 것을 원칙으로 분담금을 산출함.

: 분담금에서 KBS 현물투자비용을 제외한 나머지 금액을 KBS가 ㈜넷브로 제공함.

* + - 1. 전체 연구개발 비용 : 387,616,270원
      2. KBS 현금분담금 : 120,000,000원

: 공동연구개발 총비용(387,616,270원)/2 - KBS 현물투자비용 (73,120,270원)

(업체 공동연구개발비 총액을 양분하고 KBS 현물투자비용을 제함)

## 공동 개발 연구비 상세내역

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 내 역 | 세 부 항 목 | | 상 세 내 역 | 금 액(원) |
| 직접  인건비 | KBS | 특급 | 7,901,502원×8월×0.4명 | 25,284,806 |
| 고급 | - |  |
| 초급 | 3,972,654원×8월×0.3명 | 9,534,370 |
| 소계 | | 34,819,176 |
| 공동  개발사 | 특급 | 6,600,000원x8월x1.3명 | 68,640,000 |
| 고급 | - |  |
| 중급 | - |  |
| 초급 | 3,900,000원×8월×2.6명 | 81,120,000 |
| 소계 | | 149,760,000 |
| 합계 | | | 184,579,176 |
| 제경비 | KBS | | 직접인건비 × 110 % | 38,301,094 |
| 공동개발사 | | 직접인건비 × 110 % | 164,736,000 |
| 합계 | |  | 203,037,094 |
| 개발비 | KBS | | 직접인건비+제경비 | 73,120,270 |
| 공동개발사 | | 직접인건비+제경비 | 314,496,000 |
| 개발비합계 | | KBS개발비 + 공동개발사 개발비 | 387,616,270 |
| 각사  분담금 | KBS | | 총개발비 / 2 | 193,808,135 |
| 공동개발사 | | 총개발비 / 2 | 193,808,135 |
| KBS 현물투자비용 | | | KBS 개발비 | 73,120,270 |
| KBS 현금분담금  (부가세별도) | | | KBS 분담금 - KBS 현물투자비용 | 120,687,865 |
| KBS 현금분담금(십만 단위 절사) | 120,000,000 |

## 참여 인력 및 담당업무

|  |  |  |  |
| --- | --- | --- | --- |
| 성명 | 기술 등급 | 참여율(%) | 담당업무 |
|
| 이진원 | 특급 | 40 | 개발 총괄, 모듈별 설계, 외부 제어 기능 개발 |
| 조동현 | 특급 | 90 | 출력 기능 개발 |
| 한세희 | 초급 | 90 | 입력 기능 개발, 관리 S/W 개발 |
| 빙승재 | 초급 | 60 | 관리 S/W 개발, 출력 기능 개발 |
| 박형철 | 초급 | 60 | 관리 S/W 개발, 통합실험 |
| 문보겸 | 초급 | 50 | 외부 제어 기능 개발, 통합 실험 |