

발 간 등 록 번 호
11-1613436-000209-01

측지 VLBI 관측 시스템 소프트웨어 매뉴얼

2019. 12.



제 출 문

국토교통부 국토지리정보원장 귀하

귀 원에서 연구 위탁한 「측지 VLBI 관측 시스템 소프트웨어 매뉴얼」을 과업지시서에 따라 제출합니다.

2019. 12.

한국천문연구원 원장 이 형 목

참 여 진

참여연구진

연구책임자	한국천문연구원	변 도 영
연 구 원	한국천문연구원	위 석 오
연 구 원	한국천문연구원	신 재 식
연 구 원	한국천문연구원	윤 영 주

자문위원

서 울 대 학 교	교 수	박 용 선
한국항공우주연구원	연구원	박 덕 종
(주) 에스이티	대표이사	김 정 훈
(주) 하이게인안테나	상 무	배 민 수
(주) 하이게인안테나	이 사	손 영 득
(주) 하이게인안테나	부 장	최 경 진
(주) 스페이스웨어	부 장	김 태 형

국토지리정보원

국토지리정보원	국토측량과장	박 근 복
국토지리정보원	국토측량과 우주측지관측센터	이 상 오
국토지리정보원	국토측량과 우주측지관측센터	한 상 철

<차례>

I. KVGCS 소프트웨어 운영 매뉴얼	1
1. KVGCS 소프트웨어 구성	1
2. 관측 전 시스템 설정	3
3. 상태 확인 및 관측 제어 창	21
4. 표준화 및 지향점 관측	23
5. 본 관측 실행	27
6. CLASS를 이용한 단일경 자료 처리	29
II. wxKVG GUI 소프트웨어 매뉴얼	30
1. wxKVG 창 구성 및 개요	30
2. General 창	31
3. 수신기 제어 및 상태 창	33
4. DBBC3 창	34
5. GSM 창	35
6. Mark6 상태 창	36
7. Vex 제어 및 상태 창	38
8. Obs Control 창	40
9. Obs Tool 창	41
10. 관측 환경	44
III. VLBI 관측 매뉴얼	45
1. VLBI 관측 일정 파일 확인	45
2. Vex GUI 창을 이용한 관측 선택 및 실행	49
3. 관측 실행 전/ 진행 중 확인 내용	50
4. 관측 실행 중 로그 및 메시지 확인	51
5. 2개 이상의 연속 관측	53
IV. 안테나 캘리브레이션 라이브러리 매뉴얼	55
1. 라이브러리 구조	55
2. 안테나 구경 효율 계산 프로그램	56
3. 대기 투과율 계산 프로그램	58
4. 수신기 이득 안정도 측정 프로그램	60

5. 전체 좌표계산 라이브러리	62
부록 1. KVGCS 소프트웨어 설치 및 설정 파일	71
부록 2. DBBC3 운영 매뉴얼	76
부록 3. 이메일 이벤트 알림 설정 매뉴얼	81
부록 4. Station Log 양식	90

I. KVGCS 소프트웨어 운영 매뉴얼

1. KVGCS 소프트웨어 구성

전파 망원경을 이용하여 관측을 수행하기 위해서는 안테나, 수신기, 백엔드 시스템 등을 관측 목적에 맞게 제어해야 하고 전파 관측 자료를 처리해서 자료 파일로 저장해야 한다.

각 장치를 제어하기 위한 개별 제어 소프트웨어가 필요하며 각 장치의 물리적 위치에 따라 장치를 제어하는 소프트웨어도 서로 다른 분산된 컴퓨터에서 동작한다. 따라서 각 장치를 제어하는 제어 프로그램들과 더불어 분산되어있는 시스템들을 통합해서 제어하고 감시하는 주 관측 프로그램과 전파 수신 장치의 자료를 처리해서 자료 파일로 만들고 표시해주는 자료처리 프로그램이 필요하다. 각 프로그램 별 역할을 정리하면 다음과 같다.

- AntennaLoc : 안테나 제어 프로그램
- mcTPDLoc : 연속파 세기 측정장치 제어 프로그램
- kfft-spec : GPU 분광기 제어 프로그램
- AWSLoc : 기상 관측 시스템으로부터 기상 자료를 획득하고 저장하는 프로그램
- DSLoc : 자료 처리를 처리하고 저장하는 Data Server 프로그램
- KVGCS : 주 관측 프로그램
- wxKVG : 관측자 그래픽 환경 프로그램

위 프로그램 중 AntennaLoc, mcTPDLoc, kfft-spec, AWSLoc는 장치 제어 서버 프로그램에 해당한다. DSLoc와 KVGCS는 이들 장치 제어 프로그램의 클라이언트 프로그램이다. KVGCS의 DSLoc의 클라이언트 프로그램이며 wxKVG는 KVGCS의 클라이언트 프로그램이다. 서버와 클라이언트의 관계를 정리하면 그림1과 같다.

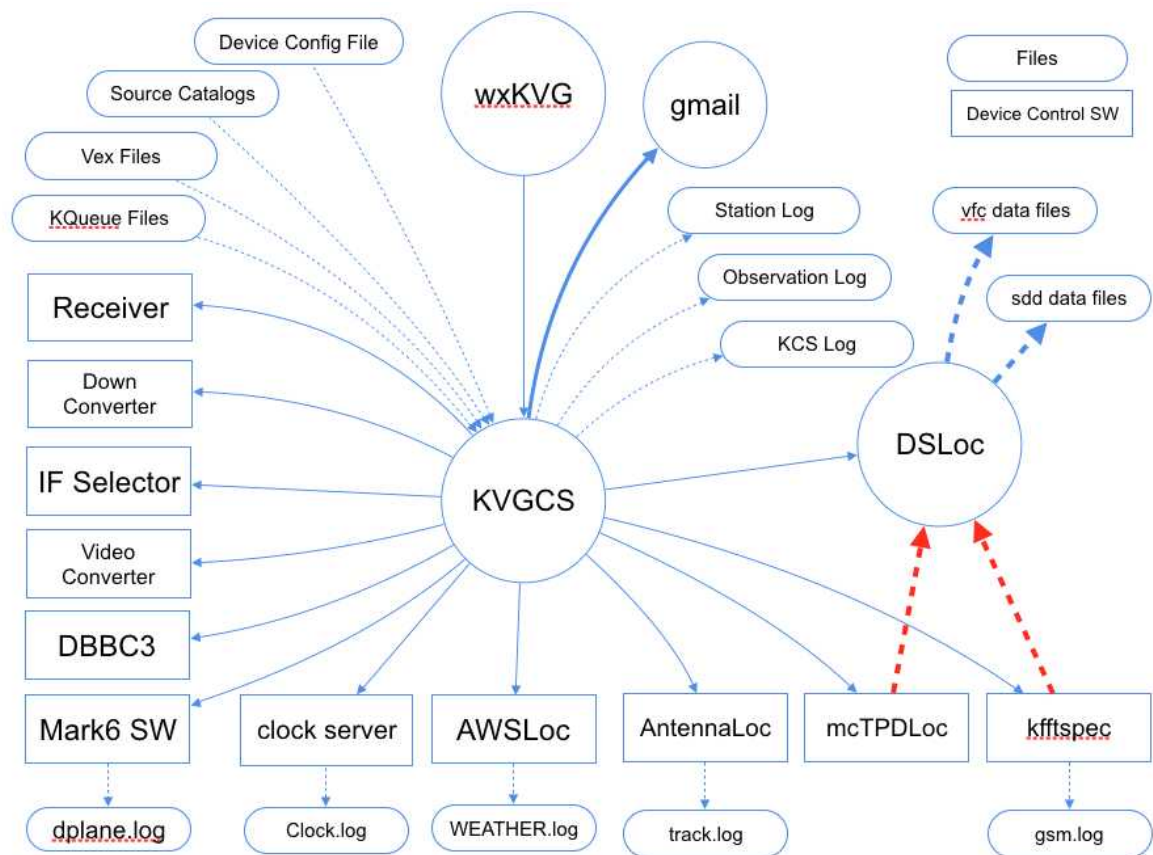


그림 1 프로그램간의 서버-클라이언트 연결 관계도

위 그림에서 사각형은 장치제어 프로그램을 의미하고 둥근 모서리 상자는 파일을 나타낸다. 관측의 주요 프로세스인 주 관측 프로그램 KVGCS, 자료 처리 프로그램 DSLoc, wxKVG 프로그램은 원으로 표시하였다. 실선으로 연결된 화살표는 제어/감시를 위한 통신의 흐름을 나타내는 데 Client-Server 역할 중 화살표의 시작점이 Client, 종료점이 Server 역할을 각각 수행한다. 각 프로그램에서 점선이 나가는 화살표는 생성되는 로그 및 자료 파일을 나타내고 프로그램으로 들어가는 점선 화살표는 보조 입력 파일을 나타낸다. 붉은 색 점선 화살표는 Data Stream의 방향을 나타낸다.

일반적으로 프로그램의 실행 순서는 Server 프로그램을 먼저 실행시키고 그 뒤에 Client 프로그램을 실행시킨다.

2. 관측 전 시스템 설정

다음은 관측에 필요한 프로세스들을 시작 방법과 관측 방법을 기술한다.

가. 소프트웨어 실행

1) 기상 상태 모니터링 프로그램 (AWSLoc) 실행

주 관측 프로그램에서는 측정 신호의 표준화, 안테나 지향 방향의 보정을 위해 온도, 습도, 기압 값을 이용한다. AWSLoc 프로그램은 주 관측프로그램에 기상 정보를 제공해 준다.

AWSLoc 프로그램을 시작시키는 방법은 다음과 같다.

```
KVGCS:~/$ cd /home/kvg/KVGCS/AWS
KVGCS:~/home/kvg/KVGCS/AWS$ StartAWSLoc
```

AWSLoc 프로그램의 동작상태를 tailw 명령을 이용하여 확인한다.

2) 안테나 제어 소프트웨어 실행

안테나 제어 소프트웨어를 실행하기 전에 FS9 소프트웨어를 중단시킨 뒤 다음 명령으로 AntennaLoc 프로그램을 시작시킨다.

```
KVGCS:~/$ cd /home/kvg/KVGCS/Antenna # 축약어 cda
KVGCS:/home/kvg/KVGCS/Antenna$ StartAntennaLoc
```

StartAntennaLoc를 실행시키면 AntennaLoc 라는 제목을 갖는 터미널이 열린다. 기본 설정으로 track.py를 메인 쓰레드로 실행시키며 천체 추적 모드를 HALT 모드로 초기화한다. track.py에서는 KVG_PMODEL.par에 있는 모델 파라미터를 읽어서 안테나 지향 위치를 보정한다.

3) 연속파 측정 소프트웨어 실행

VFC 연속파 측정 장치는 Clock 서버에서 제어하도록 설치되어 있다. Clock 서버에 접속해서 연속파 측정 소프트웨어를 실행 시킨다.

```
KVGCS:~/$ ssh clock
ClockServer:~/$ cde
ClockServer:~/KVGCS/Backend/tpd$ StartTPDLoc
```

다음 명령으로 동작상태를 확인한다.

```
ClockServer:~/KVGCS/Backend/tpd$ ps -ef | grep mcTPDLoc
kvg      19956      1  4 23:29 pts/3    00:00:02 python ./mcTPDLoc.py
```

4) 전파분광기 제어 소프트웨어 실행

가) GPU Spectrometer (GSM) 실행

GSM 제어 소프트웨어를 실행시키기 위해서는 GPU 서버에 ssh로 접속하여 해당 프로그램을 실행시킨다.

```
oper@KVG:~$ ssh gsm
difx@evlbi:$ cd gsm-run
difx@evlbi:gsm-run$ StartGSM           # GPU 분광기 소프트웨어 kfft-spec 실행
difx@evlbi:gsm-run$ tail -f gsm.log     # log 출력
```

나) Digital SpectroMeter (DSM)

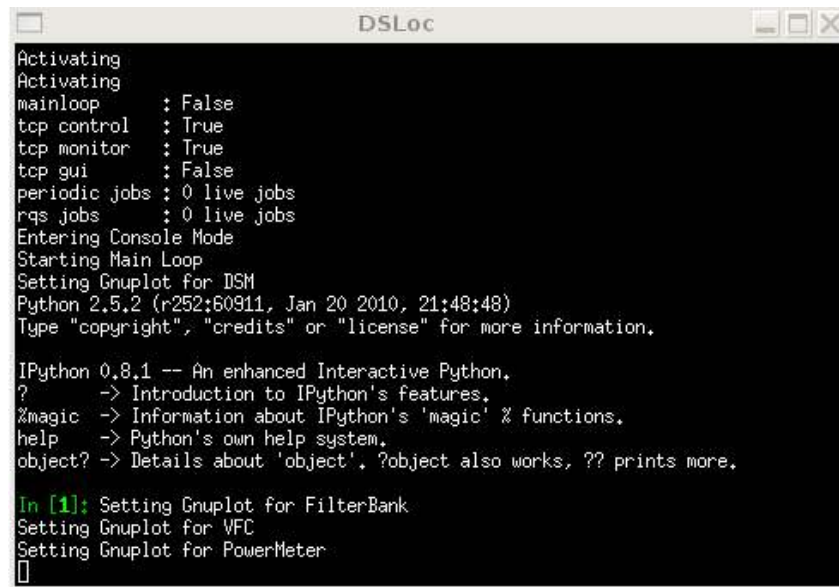
전파분광기(DSM) 제어 소프트웨어를 실행시키기 위해서는 아래와 같이 먼저 DSM에 ssh로 접속하여 해당 프로그램을 실행시킨다.

```
KVGCS:~$ ssh dsm
root@localhost# cd /home/sejong
root@localhost sejong# ./Corr.exe
```

5) Data Server 프로그램 실행

다음 명령으로 DSLoc 프로그램을 시작시킨다.

```
KVGCS:~$ cd /home/kvg/KVGCS/DataServer           # 축약어 cdd
KVGCS:/home/kvg/KVGCS/DataServer$ StartDSLoc
```



```
DSLoc
Activating
Activating
mainloop      : False
tcp control   : True
tcp monitor   : True
tcp gui       : False
periodic jobs : 0 live jobs
rqs jobs      : 0 live jobs
Entering Console Mode
Starting Main Loop
Setting Gnuplot for DSM
Python 2.5.2 (r252:60911, Jan 20 2010, 21:48:48)
Type "copyright", "credits" or "license" for more information.

IPython 0.8.1 -- An enhanced Interactive Python.
? -> Introduction to IPython's features.
%magic -> Information about IPython's 'magic' % functions.
help -> Python's own help system.
object? -> Details about 'object'. ?object also works, ?? prints more.

In [1]: Setting Gnuplot for FilterBank
Setting Gnuplot for VFC
Setting Gnuplot for PowerMeter
█
```

그림 2 자료 처리 프로그램의 터미널 창

6) KVGCS 창 실행

KVGCS:~\$ cds

KVGCS:/home/kvn/KVGCS/Main/kvncls\$ StartKVG



```
KVGCS
Begin to launch monitor thread
Start Monitoring Thread-2: monitor.py
Antenna connection closed
Rx connection closed
PtdFileServer connection closed
Begin to execute obs_common.py
DataServer connection closed
MK5B connection closed
Das connection closed
Begin to start interface server for other application
Activating
Activating
Begin to start interface server for remote control
Activating
Activating
Begin to set Scheduler
Activating
Connected by ('192.168.1.29', Ready for Operation
56885)
Adding client on fd = 42

In [1]: █
```

그림 3 KVGCS 터미널 창

KVGCS는 시작하면서 원격 장치 제어 프로그램과의 연결을 시도한다. 초기화가 끝나면 “Ready for Operation“이라는 메시지가 나온다. 중간에 Error 메시지가 나오면 그 원인에 대한 조치를 수행한다. KVGCS 시작 시 연결되는 프로그램은 장치 설정 파일에 의해 제어된다. 장치 설정 파일에 대해서는 부록. KVGCS 프로그램의 설치와 환경 설정 내용을 참고하라.

7) 관측자 GUI (wxKVG) 프로그램을 실행

KVGCS:~\$ StartGUI

위 명령을 실행하면 아래 그림 같은 창이 열리고 KVGCS 터미널에는 새로운 고객 (Client) 프로그램이 접속이 되었다는 메시지가 출력된다. KVGCS를 종료시켰다가 다시 시작시킨 경우 wxKVG 창에서 Connection 메뉴에서 Open을 선택해 주어야 KVGCS와 연결이 설정 된다. 반대로 Close를 선택하면 연결을 닫는다. 메인 창에서 각 버튼을 누르면 하위 GUI들을 열 수 있다.



그림 4 wxKVG
Main 창

나. 시스템 설정

1) 안테나 모드 설정

안테나가 천체를 추적하도록 하기 위해서는 ACU의 동작모드를 Point 모드로 바꾸어야 한다. Halt모드일 때는 wxKVG의 General 창의 Source Name영역이 빨간색으로 표시된다. 이때는 안테나를 구동할 수 없고 Point 모드 전환시켜야 한다.

안테나 제어 모드를 전환시키기 위해서는 KVGCS 터미널에서 다음 명령을 입력 한다.

```
# POINT 모드로 전환
KVGCS> antenna.point()
# HALT 모드로 전환
KVGCS> antenna.halt()
```

antenna.point() 명령을 실행시킨 뒤 wxKVG의 General 창에서 Source Name 영역이 회색으로 변하는 것을 확인한다.

2) 수신기 설정

The screenshot shows the 'Control' tab of the wxKVG receiver interface. It includes sections for 'QuasiOptics', 'Cal', 'Rx Control', and 'DownConverter Att Control'. The 'FlatMirror' and 'Chopper' are set to 'OFF'. 'PCal' and 'NoiseSource' are set to 'ON' for 2GHz and 'OFF' for 8GHz. 'Pol' and 'Mode' are set to 'RHCf' and 'SKY' for all frequencies. The 'DownConverter Att Control' section shows attenuation values for four channels across four frequencies.

그림 5 wxKVG 수신기 창의 Control 탭

수신기 선택, 편파, P-Cal, Noise Switch 등은 수신기 제어창에서 제어 할 수 있다.

FlatMirror를 제어해서 사용하고자 하는 수신기 선택

수신기 제어창에서 2/8GHz 관측의 경우 FlatMirror를 ON으로 선택한다. 반대로 22/43GHz 관측의 경우 FlatMirror를 OFF 상태가 되어야한다. 22/43GHz 관측의

관측하고자 하는 편파 선택

관측 수신 주파수에서 Pol을 선택한다.

Calibration 목적에 따라 P-Cal, Noise, Mode를 선택한다.

3) 백엔드 설정

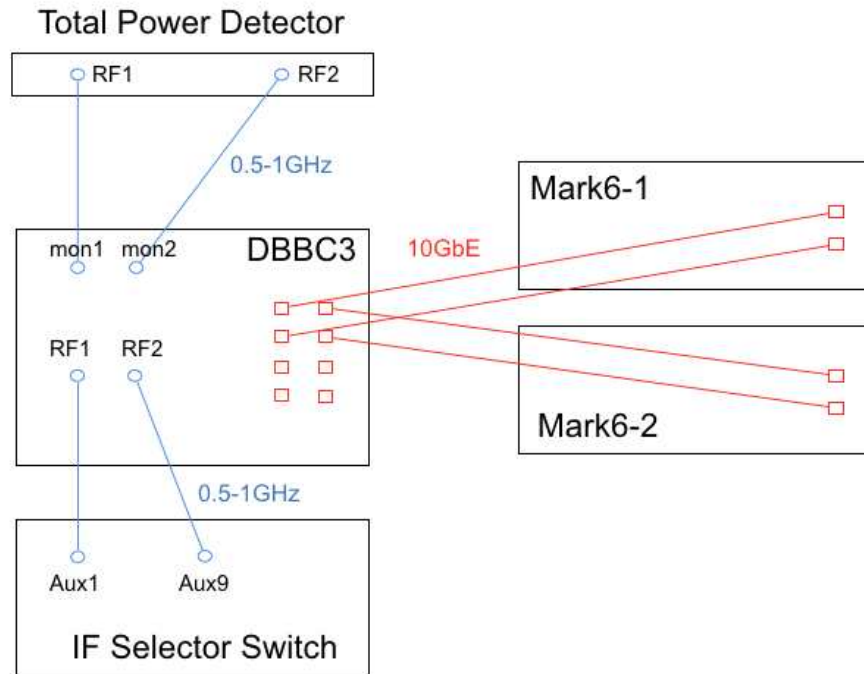


그림 6 DBBC3 입출력 신호 연결 상태

(가) DBBC3와 GSM 이용한 관측

DBBC3의 경우 DBBC 채널 주파수 설정으로 관측 주파수를 선택할 수 있다. 위 그림과 같이 IF Selector를 통해 DBBC3의 두 입력 신호가 연결된 경우 DBBC3의 GCoMO 모듈은 Down Conversion 기능은 OFF 상태가 되어야한다. IF Selector 제어을 통해 DBBC3의 입력 신호를 선택한다. IF Selector 1번과 9번에 각각 관측하고자하는 IF가 출력되도록 IF Selector 설정을 조절한다.

사용 가능한 Down Converter의 출력 IF는 총 11개로 1번부터 11번 까지 다음과 같은 주파수 범위를 가진다.

표 7 수신기 / Down Converter 입출력 주파수 범위

IF 번호	IF 이름	RF 주파수 범위 (GHz)	1 st LO (GHz)	IF 주파수 범위 (GHz)	2 nd LO (GHz)
1	S	2.0-2.5	1.50	0.50 - 1.00	-
2	X1	8.05-8.55	7.55	0.50 - 1.00	-
3	X2	8.45-8.95	7.95	0.50 - 1.00	-
4	K1	21.4-21.9	13.25	8.15 - 8.65	7.65
5	K2	21.8-22.3	13.25	8.55 - 9.05	8.05
6	K3	22.2-22.7	13.25	8.95 - 9.45	8.45
8	K4	22.6-23.1	13.25	9.35 - 9.85	8.85
8	Q1	42.26-42.76	34.11	8.15 - 8.65	7.65
9	Q2	42.66-43.16	34.11	8.55 - 9.05	8.05
10	Q3	43.06-43.56	34.11	8.95 - 9.45	8.45
11	Q4	43.46-43.96	34.11	9.35 - 9.05	8.85

예를 들어 IF S와 X2를 관측하기 위해서는 KVGCS에서 아래 명령어를 실행시킨다.

IF selector에서 입출력 선택

```
ifs.setOUT2IN(ch1 = 1)
ifs.setOUT2IN(ch9 = 3)
```

신호 세기를 조정하기 위해서는 아래 명령을 사용한다. 예를 들어 10dB attenuation level을 줄 경우는 다음과 같다.

IF selector의 출력 신호 세기 조절

```
ifs.setATT(ch1 = 10.0)
ifs.setATT(ch9 = 10.0)
```

위와 같은 DownConverter와 IF Selector를 사용한 관측 모드는 이 후 시스템의 변경에 따라 수정이 필요하다.

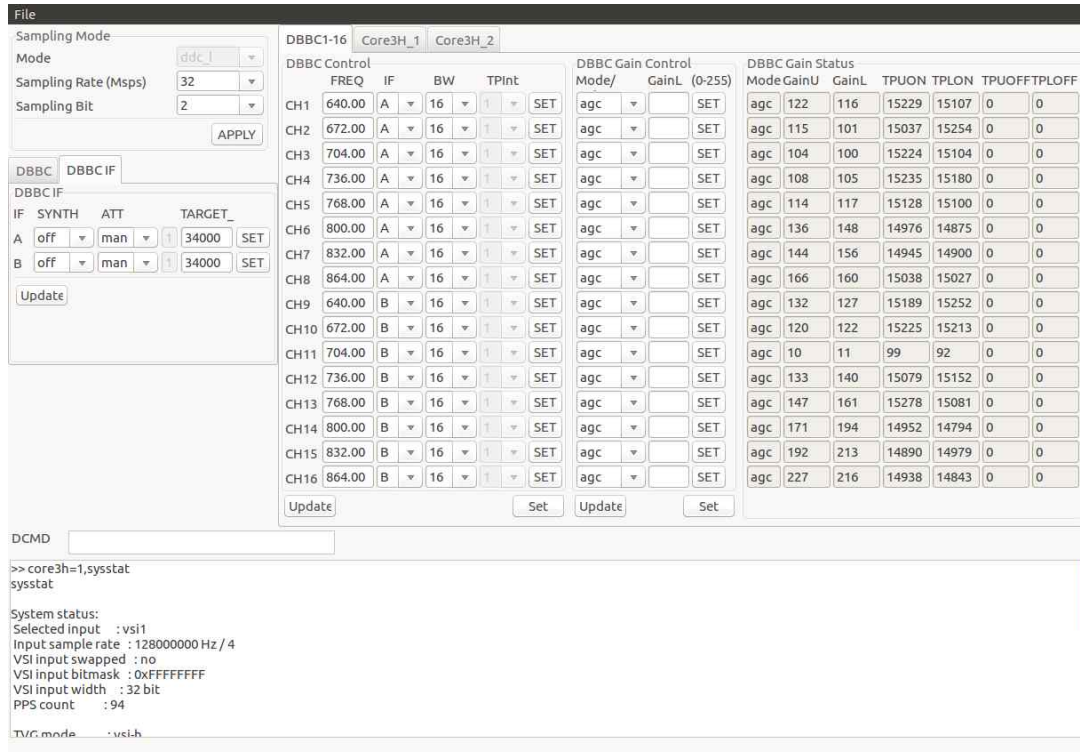


그림 7 wxKVG의 DBBC3 창 - DBBC IF 설정 및 DBBC 채널 창

DBBC3 입력 주파수 대역 선택에 따른 조절

DBBC3는 0-4GHz 대역 입력 포트와 4-15GHz 입력 포트가 있다. 4-15GHz 신호 입력은 주파수 Down Conversion 기능을 사용해서 0-4GHz로 신호의 주파수 대역을 낮춰야 한다. 이 경우 DBBC3 GCoMo 모듈의 Synthesizer를 사용하도록 enable 시켜야한다. 반대로 0-4GHz의 경우 disable 시킨다. Synthesizer 기능을 켜고 끄는 것은 DBBC3 창의 DBBC IF 패널에서 SYNTH의 모드를 선택해서 조절할 수 있다. GCoMo는 신호의 세기를 조절할 수 있다. GCoMo의 출력 신호는 DBBC3 내의 ADB3L 샘플러 보드로 입력되지만 동시에 monitor 포트도 출력된다. 이 모니터 포트 출력은 현재 Total Power Detector에 입력되고 있고 이 값을 이용해서 전체 시스템의 이득 Calibration에 사용한다. 관측 중에 GCoMo 이득 조정 모드를 AGC(Auto Gain Control)로 설정하고 사용할 경우 Calibration Chopper 또는 Noise Source를 이용한 이득 Calibration을 할 수 없다. 따라서 관측 중에는 MAN(ual) 모드로 ATT 값을 선택해 놓는다. 따라서 먼저 주파수 입력 IF 선택 후 약 30초 - 1분간 ATT를 AGC로 뒤서 자동으로 신호 세기를 적절한 값으로 자동 조절하도록 한 후 다시 MAN 모드로 전환하는 것이 좋다. 이 때 Total Power Detector의 신호 값이 0.1 근처가 되면 AGC를 MAN 모드로 전환한다.

DBBC 채널 주파수, 대역폭, 이득 설정

DBBC 1-16 채널의 주파수, 대역폭을 설정한다. 각 채널의 필드에 값을 입력하고 오른쪽 Set 버튼을 누르면 설정이 DBBC3에 적용된다. 16채널을 개별적으로 설정하는 번거로움을 줄이기 위해 왼쪽 패널의 DBBC 탭을 사용한다.

DBBC 탭에서는 값을 변경하고자하는 채널 범위를 선택 후 주파수, 대역폭 등을 입력해서 Apply 하면 오른쪽 GUI 창의 값들이 바뀐다. 변경된 값을 확인 후 아래 Set 버튼을 누르면 DBBC1-16 GUI에 표시된 값들이 DBBC3에 적용된다. DBBC1-16 채널 각각의 신호 세기는 AGC 모드로 선택해놓으면 자동으로 최적의 이득이 16채널의 이득이 조절된다.

예를 들어 22.235GHz와 43.122GHz 관측을 위한 설정 값을 정리하면 다음과 같다.

```
rx.setFloatMirror( 'Off' )  
rx.setChopper( 'Off' )
```

22.235, 43.122GHz 신호는 각각 K3와 Q3에 들어오므로

```
ifs.setOUT2IN(ch1 = 6)           # K3 (IF=6: RF=22.2-22.7) -> DBBC IFA  
ifs.setOUT2IN(ch9 = 10)          # Q3 (IF=10: RF=43.06-43.56) -> DBBC IFB  
ifs.setATT(ch1 = 10)  
ifs.setATT(ch9 = 10)
```

Synthesizer = disable, AGC로 VFC에 필요한 이득 자동 조정

```
dbbc3.setDBBCIFA(synth_oen = 0, att = 'agc', filter = 1, target_agc = 34000)  
dbbc3.setDBBCIFB(synth_oen = 0, att = 'agc', filter = 1, target_agc = 34000)
```

20초간 조정 중 기다림.

```
time.sleep(20)
```

AGC 끄고 MANUAL 모드로 전환 -> 현 Attenuation 값으로 고정됨.

```
dbbc3.setDBBCIFA(synth_oen = 0, att = 'man', filter = 1, target_agc = 34000)  
dbbc3.setDBBCIFB(synth_oen = 0, att = 'man', filter = 1, target_agc = 34000)
```

22.235GHz는 K3 채널 535MHz에, K3

```
dbbc3.setDBBC01(freq=519, channel = 'a', bw = 32, tpint =1 )   # Auto 1
```

43.122GHz는 Q3 채널 535MHz에

dbbc3.setDBBC09(freq=546, channel = 'b' , bw = 32, tpint =1) # Auto 1

위 설정으로 Core3H_1에서 출력가능한 16개 채널 중 Auto1와 Core3H_2에서 출력 가능한 16개 채널 중 Auto1이 각각 원하는 라인을 보여주게 된다.

DBBC3를 32GHz 대역폭 설정을 GUI하는 간단한 방법은 다음과 같다.

- ① Sampling Mode 패널에서 64Msps 모드를 선택해서 적용
- ② DBBC 패널에서 원하는 주파수 적용 혹은 개별 채널의 주파수 편집
- ③ DBBC1-16창 하단의 Set 버튼 클릭
- ④ Core3H_1,2 창 가운데 있는 VDIFTX의 STOP
- ⑤ Core3H_1,2의 하단 Set 버튼 클릭
- ⑥ TimeSync 버튼 클릭
- ⑦ (수정 필요 시) 네트워크 설정 편집
- ⑧ VDIFTX Start

현재 GSM 스펙트로미터는 1개의 10GbE 신호만 받으므로 2개의 Core3H 신호를 받지는 못한다. 2개의 Core3H 출력을 수신하도록 하기 위해서는 네트워크 스위치를 이용해야한다. 2개의 Core3H 중 하나를 GPU 스펙트로미터로 출력하도록 설정한다. 이 때 Core3H 창의 네트워크 설정 기능을 이용한다. 자료 전송을 위한 10GbE 사설망 구성은 다음과 같다.

Core3H_1	Core3H_2	Mark6-1	Mark6-2	GSM
10.10.1.5	10.10.1.22	10.10.1.10	10.10.1.27	10.10.1.9 (x)
10.10.1.21	10.10.1.6	10.10.1.26	10.10.1.11	10.10.1.25
10.10.1.37	10.10.1.54	10.10.1.42	10.10.1.59 (x)	
10.10.1.53	10.10.1.38	10.10.1.58 (x)	10.10.1.43	

(x) 표는 현재 비활성화된 상태임.

Netmask는 255.255.255.240 / 28로 설정해서 16개의 IP를 하나의 서브네트워크로 구성한다. 위 네트워크 설정에 맞게 DBBC3의 GUI 창에서 네트워크 출력을 설정한다.

The screenshot displays the Core3H GUI with several configuration panels:

- Sampling Mode:** Mode (ddc_1), Sampling Rate (32 Msps), Sampling Bit (2). Includes an APPLY button.
- DBBC CONFIG:** DBBC (DBBC IF), CHANNEL (1:17:1), BAND WIDTH, GAIN, FREQ START, FREQ STEP. Includes APPLY buttons.
- TIME / SYNC:** Date Time (2019-12-09T09:39:53), 1PPS Count (8155), synced status, sys dropdown, Sync button.
- VDI FTX:** stopped status, Start/Stop buttons, VSI Input (vsi1), Input Select, Sample Rate (128000000), Decimation (2), Split Mode (off), VDI F Frame (BitWidth: 2, Num of Channels: 1, Payload Size: 8024, Corner Turning: off). Includes Update and Set buttons.
- XG Destination:** Table with 4 rows showing Index, IP Address, and Thread.
- XG Arp Config:** Table with 4 rows showing Dev (eth0-eth3), Index, and MAC Address.
- XG Config:** Table with 4 rows showing Dev, IP, Gateway, and Netmask.
- DCMD:** A text input field at the bottom.

그림 8 Core3H 제어 및 감시 창

DBBC3 창에 있는 대부분의 값들은 Update 버튼을 눌러야 현재 상태에 대한 정보를 가져온다. 예외적으로 시각, 동기화 상태, VDIF 스트림 출력 상태만 자동으로 갱신된다.

VDIF 스트림 출력이 GSM로 가도록 Core3H GUI 창에서 설정을 한 뒤 GSM 창에서 원하는 스펙트럼 IF 채널을 선택하고 적당한 적분 시간과 FFT 채널 개수를 선택해 스펙트럼을 DSLoc에 출력시킨다.

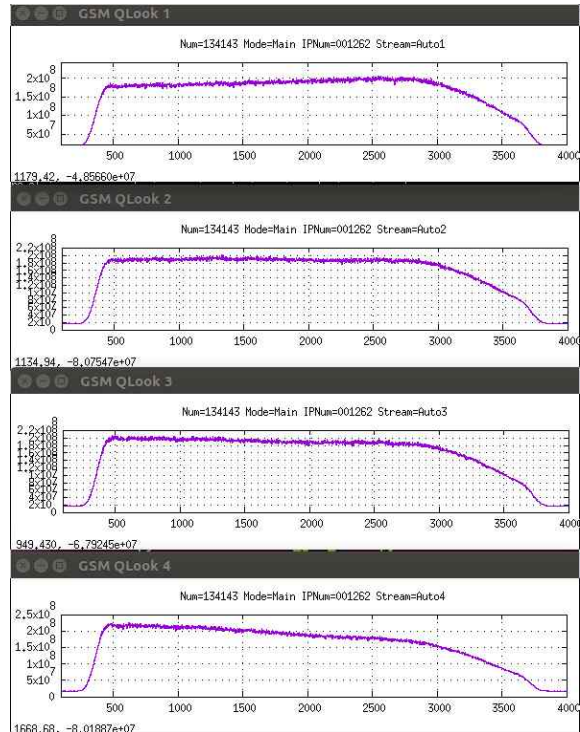
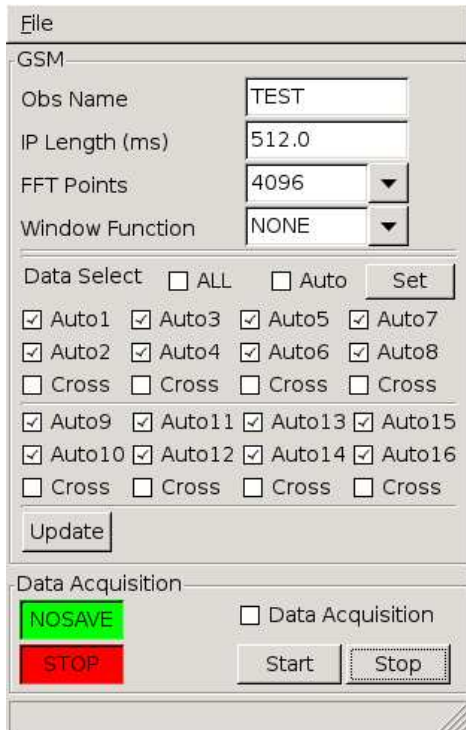


그림 9 GPU 분광기 제어 창(좌)과 GSM 출력 결과 실시간 표출 Plot(우)

DSLoc에서 gsm 자료를 받기 위해서는 dsm 쓰레드에서 신호 처리하는 스크립트를 rcv_dsm.py 대신 rcv_gsm.py로 교체한다.

```
# DSLoc 터미널
>>> dsm.stop()
>>> dsm.run('rcv_gsm.py')
```

ObsTool 창에서 Qlook Stream을 선택해서 실시간 Plot을 띄워볼 수 있다. ObsTool 창을 통한 명령은 한 번에 한 채널의 plot 만 허용하는데 위의 그림과 같이 여러 채널을 동시에 보기 위해서는 DSLoc에서 다음과 같이 입력한다.

```
>>> dsm.qlook_stream.append('Auto1')
>>> dsm.qlook_stream.append('Auto2')
>>> dsm.qlook_stream.append('Auto3')
>>> dsm.qlook_stream.append('Auto4')
```

위와 같은 방식으로 자주 사용하는 관측 세업에 대해 스크립트를 만들어 놓고 사용할 수 있다.

(나) IF Selector와 Video Converter, DSM을 이용한 관측

기존 IF Selector와 Video Converter를 이용한 관측의 경우 Video Converter의 출력 중 하나를 DSM 입력으로 연결해야한다. 일반적으로 포인팅을 확인할 경우 22GHz Water 메이저 라인 또는 43GHz SiO 메이저 라인을 사용하기 때문에 다음 내용을 이 두 라인을 관측하기 위한 설정에 대해 기술한다.

현 버전에서는 IF Selector와 Video Converter의 설정을 돕는 GUI 환경은 포함되지 않았다. 따라서 두 장비의 제어는 Command Line Interface를 통해서 하거나 혹은 장치 앞에서 직접 버튼을 눌러 입력해야한다. KVGCS 는 GUI는 제공하지 않지만 관측 스크립트에서 일괄적으로 수정하여 사용한다.

주파수 설정을 위해서는 IF Selector와 Video Converter 설정 configKVG_KQ_LSB.py 라는 Python script를 KCS에서 실행시키면 설정이 완료된다.

```
KVGCS> execfile( 'configKVG_KQ_LSB.py' )
```

다음은 configKVG_KQ_LSB.py의 내용이다.

```
#Frequency Setting for 22.235
#1st LO = 13.25
#IF for K3 = 8.95 - 9.45 (LO = 8.45)
#LO of Video Converter = 22235-13250-8450-16 = 519
```

```
ifs.setOUT2IN(ch1 = 6)
ifs.setATT(ch1 = 3.0)
vc1.setFRQ(ch1 = 519)
vc1.setATT(ch1 = 7.0)
```

```
bbc1 = rx._status.BBC1
bbc1.rf = 22.235
bbc1.ifc = 0.016
bbc1.lo = 13.25
bbc1.doppler ='Off'
```

```
#Frequency Setting for 43.122
#1st LO = 34.11
#IF for Q3 = 8.95 - 9.45 (LO = 8.45)
#LO of Video Converter = 43122-34110-8450-16 = 546
#DSM should be connected to CH9
```

```
ifs.setOUT2IN(ch9 = 10)
ifs.setATT(ch9 = 3.0)
```

```
vc2.setFRQ(ch1 = 546)
```

```
vc2.setATT(ch1 = 7.0)
```

```
bbc2 = rx._status.BBC2  
bbc2.rf = 43.122  
bbc2.ifc = 0.016  
bbc2.lo = 34.11  
bbc2.doppler = 'Off'
```

위 스크립트는 Video Converter의 1번과 9번 채널에 22GHz Water Maser 라인과 43GHz SiO 라인이 들어오도록 만들어 준다. VFC 값에 따라서 Attenuation 값을 조정해 주어야 한다.

```
# IFSelector의 9번 채널의 ATT 값 조정  
KVGCS> ifs.setATT(ch1= 5)
```

```
# Video Converter의 ATT 값 조정  
KVGCS> vc1.setATT(ch1=7)# Video Converter 1  
KVGCS> vc2.setATT(ch1=7)# Video Converter 2
```

3) 천체 선택

ObsTool 창의 Cata 탭과 Source 탭은 천체를 선택하고 안테나를 천체로 이동시킬 때, 또는 안테나의 추적 좌표에 Offset 값을 입력할 때 사용한다.

Cata 탭에서 천체 목록 파일을 선택하고 그 파일 안에 기록되어있는 천체를 선택할 수 있다. 천체 목록 파일의 양식은 Gildas 소프트웨어 꾸러미의 Astro 프로그램에서 사용하는 양식과 동일하다. 천체 목록에 기록된 천체의 이름은 모두 대문자로 통일시켜 놓는 것이 나중에 CLASS 자료 처리 프로그램에서 해당 천체의 자료를 찾을 때 편리하다.

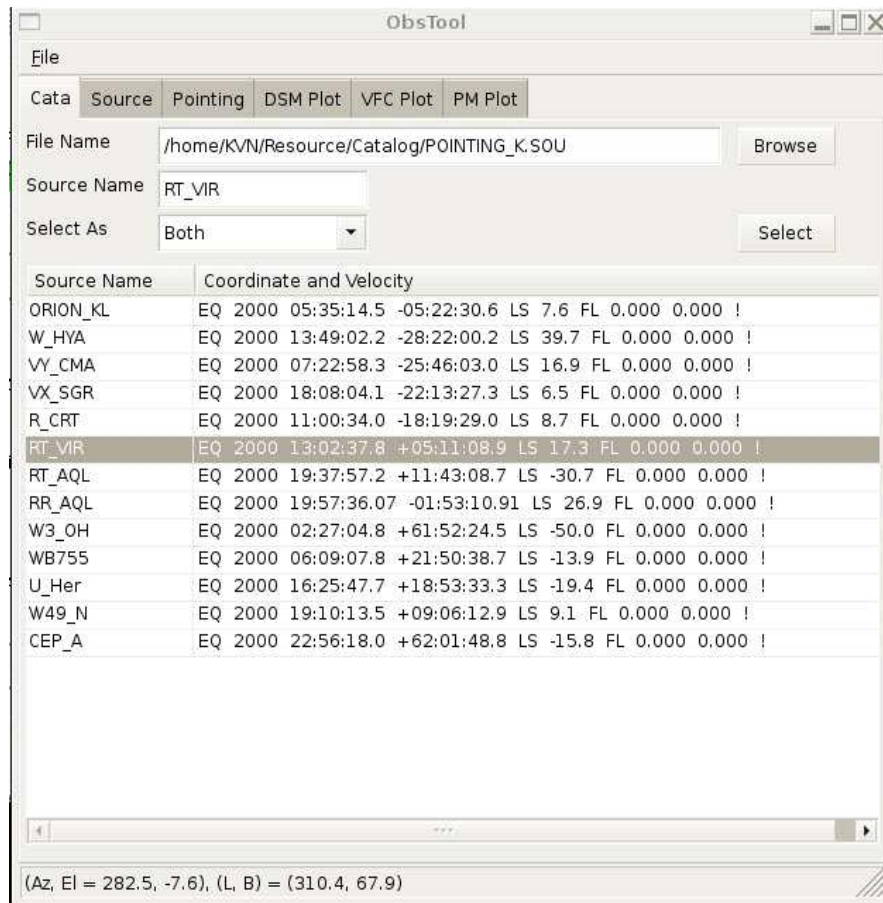


그림 10 ObsTool 창의 천체 목록 탭

Cata 탭에 표시된 천체의 행을 선택하고 Select 버튼을 누르면 선택된 천체의 좌표 값이 Source 탭 안의 좌표로 표시된다. 천체 목록에서 천체 좌표와 비교 위치 좌표를 각각 다른 좌표로 선택하려면 Select As 항목에서 Main, Ref 중에서 선택하고 천체와 비교 위치에 동일한 좌표를 적용할 경우에는 Both를 선택한다. 천체와 비교 위치에 동일한 좌표를 선택하더라도 Source 탭에서 Sky Offset 값을 다르게 적용하면 서로 다른 하늘을 바라보게 된다. Cata 창에서 천체를 선택하면 Cata 창 아래쪽에 천체의 현재 고도와 방위각, 은하 좌표계가 표시된다.

* 주의: 안테나가 Halt 상태가 되면 자동으로 Reference 위치의 Sky Offset 값이 0으로 변한다. 따라서 Point 후에는 새로 Offset 값을 입력해주어야 한다.

주로 사용하는 Catalog 디렉토리는 /home/kvg/KVGCS/Resource/Catalog 이다. 이 디렉토리에 있는 공통으로 쓰이는 Catalog 파일은 다음과 같다.

- POINTING_K.SOU : 22GHz 물 분자 메이저 방출 천체 목록

- POINTING_Q.SOU : 43GHz SiO 메이저 방출 천체 목록
- CONTINUUM_K.SOU : 22GHz에서 밝은 연속전파 방출 천체 목록
- CONTINUUM_Q.SOU : 43GHz에서 밝은 연속전파 방출 천체 목록
- POL.SOU : 연속파 편광 천체 목록

4) 천체 추적

The screenshot shows the ObsTool application window with the 'Source' and 'Sky' tabs selected. The 'Source' tab has the following fields: Planet (None), SourceName (4C39.25), CoordSys (Eq), Epoch (2000), Vel (km/s) (0), and VelSys (Lsr). The 'Sky' tab has the following fields: Position (Main), RaOffsetSys (Sky), ZoneMode (Auto), and ZoneReq (0). Below these are three tables for Equatorial (Eq), Galactic (Ga), and Horizontal (Ho) coordinates and offsets for Main and Reference positions.

Eq	Ra	Dec
Main	09:27:03.01	39:02:20.9
Ref	09:27:03.01	39:02:20.9

Eq	Ra Off	Dec Off
Main	00:00:00.00	00:00:00.0
Ref	00:00:00.00	00:00:00.0

Ga	L	B
Main	183.70853676	46.163715157
Ref	183.70853676	46.163715157

Ga	L Off	B Off
Main	0.0	0.0
Ref	0.0	0.0

Ho	Az	El
Main	74.1707251455	57.0964874345
Ref	74.1707251455	57.0964874345

Ho	Az Off	El Off
Main	0.0	0.0
Ref	0.0	0.0

Buttons: Set, Update

그림 11 ObsTool의 천체 좌표 제어 창

Cata 탭을 이용해서 천체 좌표를 입력한 후 Source 탭에 표시된 좌표로 안테나를 이동하려면 아래쪽에 있는 Set 버튼을 누르면 된다.

천체와 비교 위치에서 상대적인 Offset을 적용하려면 Sky Column에 해당 좌표계에 Offset 좌표를 입력하고 Enter를 친다. Sky Offset 입력 부분에서는 천체에 대한 Offset과 비교 위치에 대한 Offset을 별도로 입력할 수 있도록 되어있다.

Sky 상자에서 Position는 Main, Ref 중 하나를 선택할 수 있는데 각각은 안테나를 천체 위치와 비교 위치로 이동시킨다. Sky에 Ra Offset을 주었을 때 그 값이 하늘에서의 각 거리를 의미하는지 아니면 좌표 값의 차이를 의미하는 지에 따라 RaOffsetSys 값을 Sky 또는 Coord를 선택한다. (* 주의: Grid 관측 시 RaOffsetSys를 Sky로 해놓는다.)

Sky 상자에 있는 ZoneMode는 안테나 방위각의 구간을 제어하는 모드를 선택하는 데 사용된다. (안테나 방위각은 -90 ~ 450도를 움직일 수 있다. 천체의 방위각이 60도인 경우 안테나 방위각을 60도 또는 420도로 움직일 수 있다.) ZoneMode는 Auto와 Manual 두 개의 모드가 있다. Auto일 경우는 제어 소프트웨어가 자동으로 방위각 구간을 제어하고

Manual일 때는 ZoneReq의 구간으로 방위각을 설정한다. ZoneReq는 -1,0,1의 세 구간이 있는데 각각 -90~0, 0~360, 360~450의 구간을 의미한다.

3. 상태 확인 및 관측 제어 창

가. 상태 확인

General 창은 현재 시각, 안테나가 지향하는 천체의 좌표와 현재 안테나의 좌표, 관측 진행 단계, 기상 상태, Backend 상태 등 관측 상황과 현재 시스템의 상태를 파악하는데 필요한 정보들을 모아놓은 창이다. General 창은 제어 기능은 없으며 단지 현재 상태 정보를 표시하기만 한다. General 창에 나타나는 시각 정보는 안테나 ACU의 시각 정보를 표시해 주는 것인데 KCS와 ACU와의 통신이 끊겨 있거나 또는 KVGCS에서 ACU의 정보를 주기적으로 읽어오는 monitor.py 쓰레드가 정지되어 있으면 시간이 진행하지 않게 된다. 또는 wxKVG와 KVGCS 사이의 연결이 종료되어 있어도 시간이 변하지 않게 된다.

관측하고자 하는 천체와 안테나의 좌표를 표시하는 창들과 더불어 ObsPgm 상자는 관측의 진행 상태를 표시해 준다. General 창의 가장 아래에 있는 상자에서는 주파수 설정과 자료의 상태를 요약해서 보여준다.

The screenshot displays the 'General' status window of the wxKVG software. It is organized into several sections:

- Antenna:** Set to 'Hexapod'.
- Time:** UT 2019-12-13(347) 05:34:57, LST 19:30:39.
- SOURCE:** R_CAS, with buttons for Limit, Ok, Position, Main, COORD, and Eq.
- Coordinates:** RA (23:58:24.869), DEC (51:23:19.700), AZ (49.77526), EL (41.27649).
- Offsets:** OFFSET (00:00:00.0), ACTUAL (23:59:25.312), L,B (0.00000), L,B OFF (0.00000), ZDES (14000).
- COMMAND:** 49.76600, 41.18400, SKY OFF (0.00000), ERROR (") (-2.17), OFFSET (") (0.00), REFRAC (") (69.71).
- ObsPgm:** ObsMode (Idle), Repeat (1 of 1), ScanNum (134113), Scan (8 of 8), Status (Main), Tint (0.0 of 0.0).
- Weather:** RAIN (NO RAIN), WIND (2.2 m/s), TEMP (5.3 C), DIRECT (347.7 deg), HUMID (39.7%), PRESS (1000.6 mbar).
- RxRoom:** TEMP (293.0 K), HUMID (30.0 %), CLOCK (-33.8 us).
- Frequency Table:**

	Rx	RF (GHz)	Vel	Dplr	PLL	Tau0	Tau	Tsys (K)	P_Sky	P_Hot	P_Cold	ADS #	BSD (%)
IF A	22L	22.235000	0.000	Off	True	0.187	0.000	537.3	0.0818	0.1230	0.0818	None	
IF B	43L	43.122000	0.000	Off	True	0.207	0.000	727.1	0.0720	0.0989	0.0720	None	

그림 12 wxKVG의 General 상태 창

나. 관측 제어 창

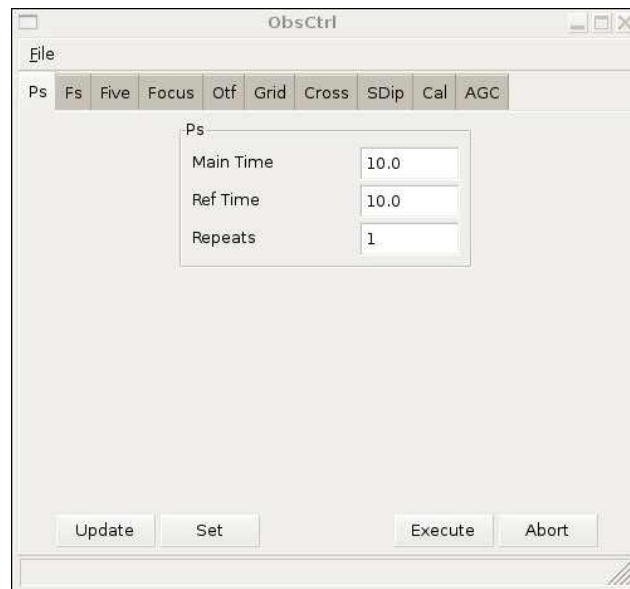


그림 13 관측 모드 제어 창

Obs Control 창에서 지원되는 각 관측 모드 탭에는 창 아래 부분에 Update, Set, Execute, Abort 버튼을 가지고 있다. 각 버튼의 기능은 다음과 같다.

- Update : 현재 KCS에 적용되어있는 파라미터 값을 가져와 표시한다.
- Set : 현재 창에 표시되어 있는 값들을 KCS에 적용한다. 혹은 개별 관측 파라미터 입력 칸에 값을 입력하고 Enter를 치면 파라미터가 KCS에 전달된다.
- Execute : 선택된 탭의 관측 모드를 실행시키는 명령을 KCS에 보낸다. 관측이 실행되면 KCS창에 관측 시작 메시지가 나타난다.
- Abort : KCS에서 진행 중인 관측을 중단시킨다. 관측이 중단되면 KCS 터미널에 관측이 강제 종료되었다는 메시지가 나타난다.

4. 표준화 및 지향점 관측

가. 표준화 관측

1) AGC 실행

AGC (Automatic Gain Control)는 샘플러와 VFC에 입력되는 신호의 세기를 자동으로 조절해 주는 모드이다. AGC로 수신기 이득을 조절하면 Cal을 해서 시스템의 이득을 측정해야 한다.

* 주의 : 안테나가 적당한 고도에 위치했을 때 실행해야 한다.

2) SkyDip 실행

SkyDip은 대기의 광학적 두께를 측정하는 데 사용한다. SkyDip 과정은 Cal을 포함하기 때문에 SkyDip을 할 때에는 Cal을 생략해도 된다.

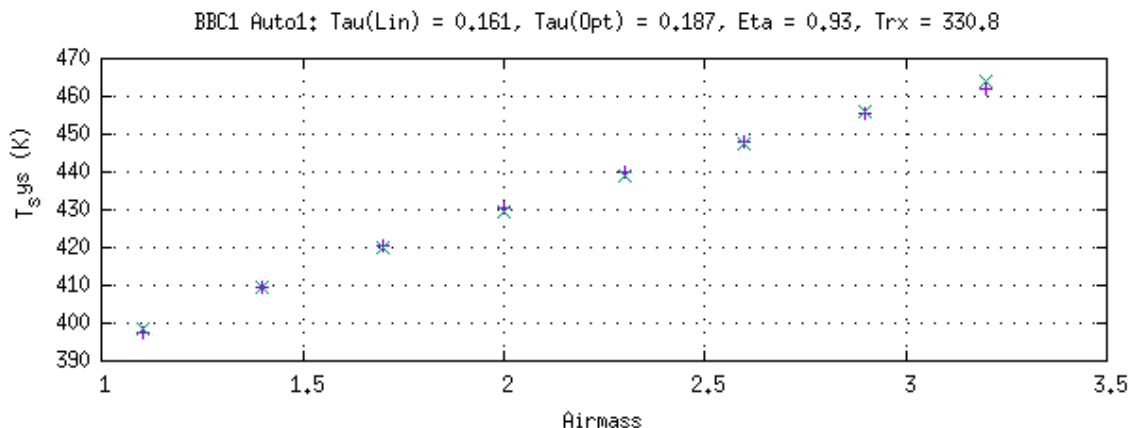


그림 14 Total Power 측정기를 이용해 측정한 SDip 결과 Plot

SkyDip을 실행시키면 DSLoc가 결과를 Plot으로 표시한다. SkyDip이나 Cal을 수행한 뒤 General 창의 맨 아래에 있는 자료 상태 상자에 있는 P_HOT 값을 확인한다. P_HOT 값은 Vane으로 가렸을 때 VFC 값에서 Blank일때의 VFC 값을 뺀 값이다. P_HOT 값이 정상 범위를 넘어서면 빨간색으로 표시되는데 이때에는 Rx Control 창에서 해당 BBC의 Att 값을 높인 후 다시 SkyDip이나 Cal을 수행한다.

나. 안테나 지향점 및 부경 초점 확인

1) 지향점 확인

Five 또는 Cross 로 안테나의 지향점을 조절한다. Five로 지향점을 확인하기 위해서는

방출선의 위치와 Base line의 위치를 설정해 주어야 한다.

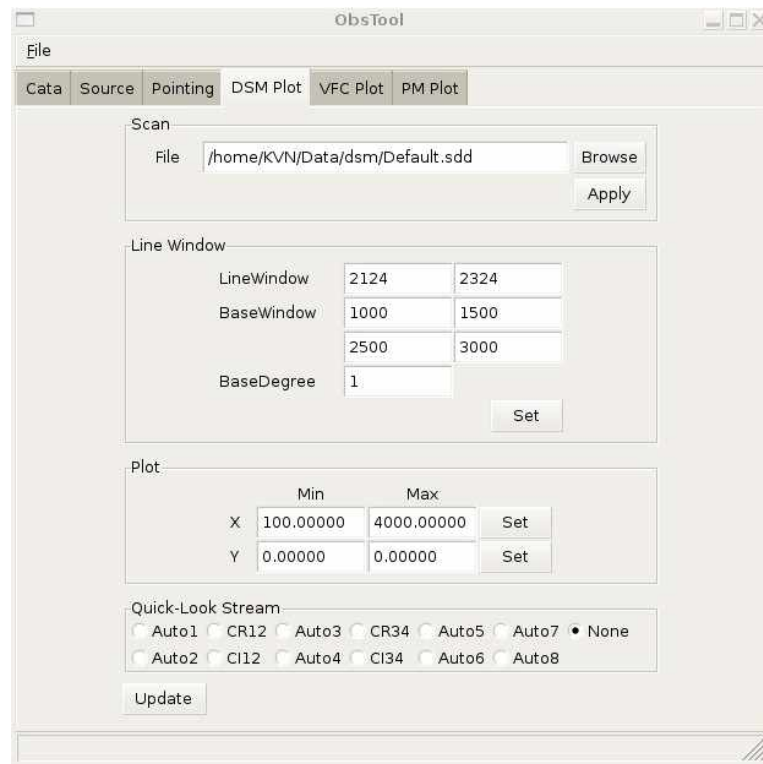


그림 15 ObsTool 창 의 DSM Plot 탭

ObsTool의 DSM Plot 탭에서 DSLoc 프로그램에서 Five나 Focus 관측 결과를 실시간 해석하기 위해 필요한 Line 영역과 baseline fitting 범위와 baseline fitting 다항 함수의 차수를 설정한다. 여러 개 채널을 동시에 관측할 경우 개별적으로 다른 값을 설정할 수 없다. 동일한 값이 여러 채널 (예: Auto1, Auto2, Auto3,...) 해석에 적용되기 때문에 지향 오프셋이나 초점을 확인하고자 하는 천이선에 대한 값을 설정한다. BaseDegree 는 다항 함수의 차수를 선택하는 값인데 -1인 경우 baseline fitting을 하지 않는다.

```

DSLoc
BBC# = 2 ADS# = 2 Source = 0.0751
Writing CLASS scan 14537 ... Done
*****
Sequence # = 5
Auto1 BBC1 Scannum:14538 IPNum:195, IPLen:102.40, Mode:Main
:: ri,wi = 29,30 > 1286938637.14 20,370 0.516

BBC# = 1 ADS# = 1 Source = 0.0620
Writing CLASS scan 14538 ... Done
*****
Scannum : [14522, 14526, 14530, 14534, 14538]
Area : [267.24, 580.30, 316.59, 318.99, 301.63]
Offset : [-1.452, -3.987]
BeamSize: [136.754, 130.234]
*****
Auto2 BBC2 Scannum:14539 IPNum:195, IPLen:102.40, Mode:Main
:: ri,wi = 29,30 > 1286938637.14 20,657 0.545

BBC# = 2 ADS# = 2 Source = 0.0752
Writing CLASS scan 14539 ... Done
*****
Scannum : [14523, 14527, 14531, 14535, 14539]
Area : [9.24, 12.08, 11.49, 11.58, 8.95]
Offset : [-24.426, -22.179]
BeamSize: [261.623, 271.586]
*****
Auto3 BBC2 Scannum:14540 IPNum:195, IPLen:102.40, Mode:Main
:: ri,wi = 29,30 > 1286938637.14 20,953 0.517

BBC# = 2 ADS# = 2 Source = 0.0752
Writing CLASS scan 14540 ... Done
*****
Scannum : [14524, 14528, 14532, 14536, 14540]
Area : [1.07, 1.15, 5.51, 3.92, 10.48]
Offset : [-9.303, 35.730]
BeamSize: [82.610, 125.326]
*****
Auto4 BBC2 Scannum:14541 IPNum:195, IPLen:102.40, Mode:Main
:: ri,wi = 29,30 > 1286938637.14 21,259 0.537

BBC# = 2 ADS# = 2 Source = 0.0752
Writing CLASS scan 14541 ... Done
*****
Scannum : [14525, 14529, 14533, 14537, 14541]
Area : [4.84, 3.58, 10.81, 5.42, 5.31]
Offset : [0.774, 18.600]
BeamSize: [170.437, 129.165]
*****
In [6]: 

```

그림 16 DSLoc 터미널 출력 예 (Five)

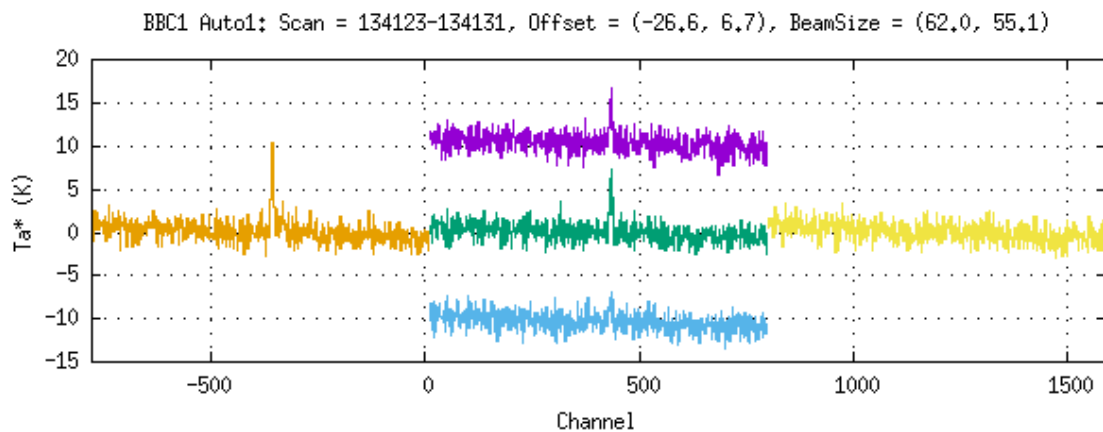


그림 17 전파분광기로 얻은 Five 관측 결과 Plot

DSLoc는 Five 관측이 끝나면 그림과 같이 관측 결과를 표시한다. 현재 Offset에서 Five 결과에 표시된 Offset 만큼 Pointing Offset을 이동시킨다. 예를 들어 현재 Offset이

(-7.0, 7.0)인데 Five 관측 결과 표시된 Offset 값이 (5.0,-3.0)이라면 Offset Input에 (-2.0,4.0)을 입력한다. ObsTool의 Pointing 탭을 이용해서 Pointing Offset과 Focus를 조절할 수 있다.

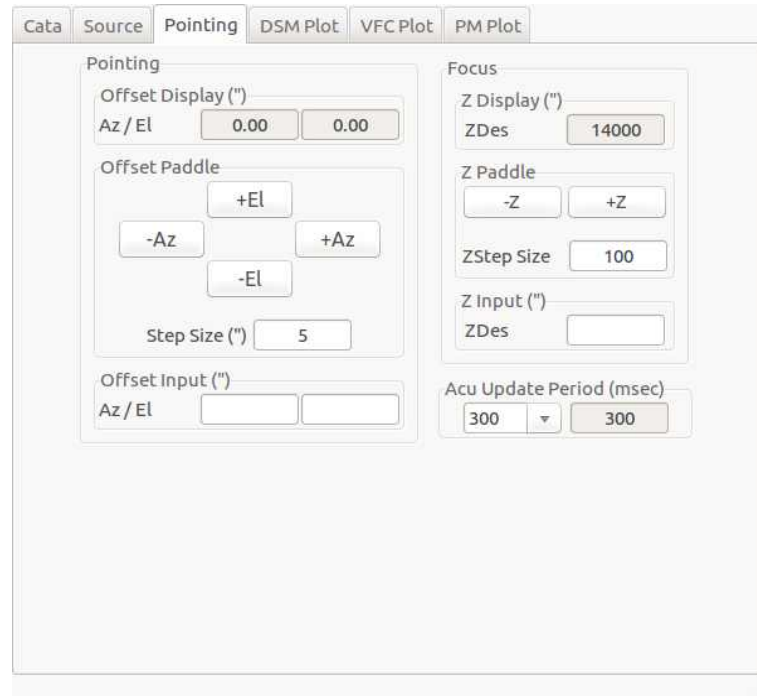


그림 18 안테나 지향 오프셋 및 초점 조절 창

2) 초점 조절

Focus를 실행해서 초점 위치를 확인한다. Focus 관측을 통해 얻은 부경의 최적 위치로 이동시키려면 ObsTool의 Pointing 탭에서 ZDes 값을 입력한다.

43GHz 주파수에서 Focus 관측을 할 때 ZStep을 2000(μm)로 놓고 관측하면 부경이 -4000, -2000, 0, 2000, 4000(μm) 위치에서 관측을 하는 데 -4000과 4000 위치에서는 신호의 세기가 절반 이상 줄어든다. 그림 16는 Focus 관측을 결과를 DSLoc가 화면에 표시하는 Plot이다. 최적 초점위치를 해석한 결과가 상단에 ZOffset 값으로 표시된다. Focus 관측을 실행할 때 설정한 Z0 값에 ZOffset 만큼 더한 위치로 부경의 위치를 이동시켜서 관측한다.

주파수 별로 초점의 위치가 다르게 나타날 수 있으므로 각 주파수 별로 초점의 차이가 얼마인지 확인해두는 것이 좋다.

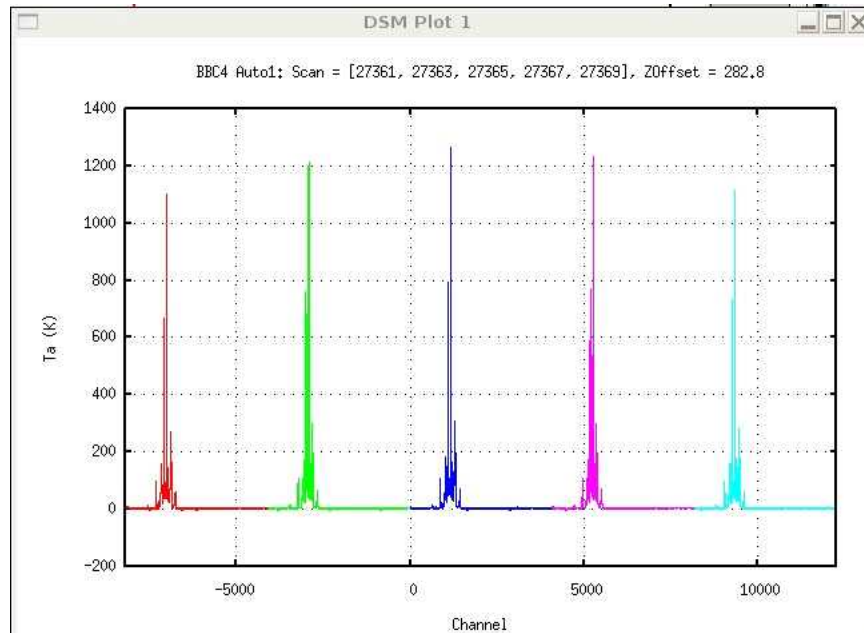


그림 19 Focus 관측 결과 표시 화면

5 본 관측 실행

관측하고자 하는 대상으로 이동해서 VLBI 관측 또는 단일경 모드를 이용한 PS, Grid 등의 본 관측을 실행한다. 대략 1-2시간 간격으로 SkyDip, Pointing, Focus 등을 확인하는 것이 바람직하다. 특히 해 뜨고 질 무렵에는 태양의 열적 효과에 의해 지향오차가 커지므로 (최대 시간당 10초) 주의해야 한다. 다음은 각 관측에 대한 간략한 설명이다.

가. VLBI 관측

VLBI 관측은 관측 종류에 따라 관측 시간이 달라지지만 보통 8시간에서 24시간의 긴 관측을 하나의 VLBI 관측 일정 파일로 수행한다. 긴 시간 관측하는 동안 시스템의 성능이 변할 수 있기 때문에 관측 중간에 다시 시스템 이득, 안테나 지향 오차 등을 측정해서 보정한다. VLBI 관측의 보다 자세한 방법은 3장에서 기술한다.

나. Position Switching(PS) 관측

PS (Position Switching) 관측은 천체와 천체 주변의 인접한 빈 하늘을 번갈아 관측하면서 두 값의 차를 얻는 모드이다. 이상적인 경우 두 지점을 관측하는 사이에 대기과 시스

템의 특성이 거의 같다면 두 값의 차는 대기와 시스템의 특성이 제거되고 관측하고자 하는 천체의 특성만 남게 된다. 하지만 실제로는 대기의 특성과 시스템의 특성이 시간에 따라 변하기 때문에 완벽히 제거되지는 않는다. 분광 관측의 경우 나중에 신호 선이 있는 지역을 제외한 나머지 영역의 특성을 내삽하여 신호선이 있는 주파수에서의 대기나 시스템의 기여분을 제거한다. 보통 천체와 비교 천체 볼 때 적분시간을 30초 이하로 설정해 놓고 관측한다. 이 시간이 길수록 그 사이에 대기와 시스템의 특성이 더 많이 변하므로 Baseline의 특성이 나빠진다. 반대로 너무 짧게 하면 안테나가 위치를 바꾸고 안정될 때까지 기다리는 시간이 전체 관측 시간 때문에 생기는 시간 손실이 많아진다.

다. 격자점(Grid) 관측

Grid (격자점) 관측은 안테나 빔 크기보다 큰 영역에 대한 전파세기의 분포를 얻기 위해 사용하는 관측 방법이다. 기본적인 원리는 PS관측과 동일하지만 비교 위치를 한 번 보고 여러 격자점을 관측한다.

6. CLASS를 이용한 단일경 자료 처리

DSLloc는 관측 자료를 Gildas 패키지의 CLASS의 자료 양식으로 만들어 Default.sdd 파일에 기록한다. CLASS 프로그램을 이용하면 새로 기록되는 스펙트럼을 표시하고 해석할 수 있다. (CLASS에서의 자료 처리 방법은 <https://www.iram.fr/IRAMFR/GILDAS/doc/html/class-html/class.html> 를 참고하라.)

```
KVNYS:~/$ cd /home/kvg/KVGCS/Data/DSM (or cddsm)
KVNYS:~/home/kvg/KVGCS/Data/DSM$ class
LAS> set mode x 2100 2300 (or sx 2100 2300)
LAS> set mode y auto (or sy auto)
LAS> file in Default.sdd          # Class 파일 열기
```

CLASS는 스크립트도 실행시킬 수 있는데 qlook.class 라는 이름의 스크립트를 수행하는 방법은 다음과 같다.

```
LAS> @qlook.class
```

앞의 '@' 부호가 스크립트 실행을 나타내는 문자열이다. qlook.class 스크립트를 동작시키면 Class 파일에 새로 추가되는 스펙트럼을 자동으로 표시한다. qlook.class 스크립트 수행 중 표시 영역을 변경하기 위한 과정은 다음과 같다.

CTRL-C로 스크립트를 멈춘뒤 quit 명령으로 LAS> 프롬프트로 돌아간다.

```
LAS> set mode x x1 x2
LAS> @qlook.class
```

관측자는 자신의 관측 설정에 따라 qlook.class를 수정해서 사용할 수 있다.

II. wxKVG GUI 소프트웨어 매뉴얼

1. wxKVG 창 구성 및 개요

GUI를 제공하는 wxKVG 프로그램은 KVGCS의 상위 Client 프로그램으로 KVGCS에서 정보를 받아 GUI 창에 표시해준다. 또한 관측자가 필요한 관측 제어도 GUI 창에서 실행 버튼을 누르면 실행 명령이 KVGCS 전달되어 해당 작업을 수행하게 된다. 이를 위해서 KVGCS는 TCP/IP 명령어 원격 인터페이스를 제공하는데 단순한 장치 제어 명령의 전달 뿐 아니라 KVGCS에 설정된 함수 또는 스크립트 등도 원격 실행이 가능하다.

다음은 wxKVG GUI 프로그램을 실행시켰을 때 열리는 창으로 창의 각 버튼은 관련 내용을 표출 또는 제어하기 위한 창이 열리는 구조로 되어있다. wxKVG의 각 하위 창의 기능을 정리하면 다음과 같다.



그림 1 wxKVG
Main 창

표 1 wxKVG GUI 하위 창의 기능

창 이름	주요 기능
General	<ul style="list-style-type: none"> 관측에 필요한 정보를 종합적으로 표시 시각, 천체 추적 상태, 기상, 캘리브레이션, 시스템 잡음, 관측 모드, 지향 오프셋, 초점 등
Obs Control	<ul style="list-style-type: none"> 관측 자료 및 안테나 성능 캘리브레이션에 필요한 관측 모드의 파라미터 설정 및 실행
Obs Tool	<ul style="list-style-type: none"> 관측 천체의 선택, 안테나 제어, 자료 표출 및 저장 방식 등 제어
Rx	<ul style="list-style-type: none"> 수신기, Down Converter의 상태 표시 및 제어
DBBC3	<ul style="list-style-type: none"> DBBC3의 제어 및 상태 표시
GSM	<ul style="list-style-type: none"> GPU 스펙트로미터의 제어 및 상태 표시
Mark6	<ul style="list-style-type: none"> Mark6 기록기의 상태 표시
Vex	<ul style="list-style-type: none"> VLBI 관측 스케줄 파일 선택 및 관측 실행, VLBI 관측 진행 상황 표출
Close All	<ul style="list-style-type: none"> 전체 하위 GUI 창을 닫는 기능

2. General 창

General 창은 시각, 추적 중인 천체의 좌표 정보, 안테나의 지향 방향 및 추적 오차, 기상 상태, 수신기 방의 온도, 시각 시스템의 Clock Offset, 관측 모드의 진행 상태 등을 표출한다. 또한 하단에는 DBBC3에 선택된 수신기, 관측 주파수, 대기 투과율 측정량, 시스템 잡음 온도, Calibration Power 정보가 표출된다. 이 값들은 모두 실시간으로 업데이트 되는데 모든 정보는 KVGCS에서 가져오는 값들이다. 현재 값들 중 주의가 필요한 값들은 노란색 또는 빨간색으로 표시되는데 안테나 추적 오차, 바람의 세기 등이 일정 크기 이상일 경우 노란색 또는 빨간색으로 표시 되며, 안테나가 천체 추적이 불가능한 Halt 또는 System Error 인 경우 천체 이름이 빨간색으로 표시된다.

Hexapod 탭에는 부경의 현재 위치 정보가 표출된다. 부경은 고주파수(22/43GHz) 관측 일 경우 Z 값이 140000 오프셋이 최적이고 2/8GHz 관측의 경우 0 값이 정상 값이다. 이 오프셋 값은 General 창의 ZDes로도 표시된다.

File

Antenna Hexapod

UT 2019-12-13(347) 05:34:57 LST 19:30:39.

SOURCE R_CAS Limit Ok Position Main COORD Eq

	RA	DEC	AZ	EL
EPOCH	23:58:24.869	51:23:19.700	COMMAND 49.77526	41.27649
OFFSET	00:00:00.0	00:00:00.0	ACTUAL 49.76600	41.18400
ACTUAL	23:59:25.312	51:30:08.445	SKY OFF 0.00000	0.00000
L,B	0.00000	0.00000	ERROR (") -2.17	-15.23
L,B OFF	0.00000	0.00000	OFFSET (") 0.00	0.00
ZDES (")		14000	REFRAC (")	69.71

ObsPgm

ObsMode Idle Repeat 1 of 1

ScanNum 134113 Scan 8 of 8

Status Main Tint 0.0 of 0.0

Weather

RAIN NO RAIN WIND 2.2 m/s

TEMP 5.3 C DIRECT 347.7 deg

HUMID 39.7 % PRESS 1000.6 mbar

RxRoom

TEMP 293.0 K

HUMID 30.0 %

CLOCK -33.8 us

	Rx	RF (GHz)	Vel	Dplr	PLL	Tau0	Tau	Tsys (K)	P_Sky	P_Hot	P_Cold	ADS #	BSD (%)
IF A	22L	22.235000	0.000	Off	True	0.187	0.000	537.3	0.0818	0.1230	0.0818	None	
IF B	43L	43.122000	0.000	Off	True	0.207	0.000	727.1	0.0720	0.0989	0.0720	None	

그림 2 wxKVG의 General 창

File

Antenna Hexapod

	X (um)	Y (um)	Z (um)	Tip (")	Tilt (")
Actual	1499	-999	14000	0	0
Desired	1500	-1000	14000	0	0
LutCor	0	0	0	0	0
Model...	0	0	0	0	0
Cmd	0	0	0	0	0

ObsPgm

ObsMode Idle Repeat 1 of 1

ScanNum 134143 Scan 5 of 5

Status Main Tint 0.0 of 0.0

Weather

RAIN NO RAIN WIND 2.0 m/s

TEMP 5.2 C DIRECT 4.4 deg

HUMID 37.0 % PRESS 1000.6 mbar

RxRoom

TEMP 293.0 K

HUMID 30.0 %

CLOCK -33.8 us

	Rx	RF (GHz)	Vel	Dplr	PLL	Tau0	Tau	Tsys (K)	P_Sky	P_Hot	P_Cold	ADS #	BSD (%)
IF A	22L	22.235000	0.000	Off	True	0.187	0.277	704.3	0.0813	0.1230	0.0818	None	
IF B	43L	43.122000	0.000	Off	True	0.207	0.306	986.3	0.0719	0.0989	0.0720	None	

그림 3 wxKVG General 창의 부경 탭

3. 수신기 제어 및 상태 창

수신기 창은 수신기와 Down Converter, P-Cal, NoiseSource, FlatMirror, Calibration Chopper, 편파 스위치의 제어와 상태 표시에 사용된다. Status 탭과 Control 탭으로 구성되어 있는데 Status 탭에는 현재 상태가 표시를 Control 탭은 제어 목적으로 사용한다. Status 탭에 표시되어 있는 값들은 실시간으로 자동 업데이트 된다.

The screenshot shows the 'Status' tab of the wxKVG receiver interface. It includes sections for QuasiOptics (FlatMirror: None, Chopper: None), Cal (P-Cal: OFF, NoiseSource: OFF), Rx Status (Pol: RHCP, Mode: SKY), DownConverter Power Level (four channels with power values like -43.84, -39.84, etc.), and RxRoom (Temp (K) and Humid (%)).

그림 4 wxKVG 수신기 창의 Status 탭

The screenshot shows the 'Control' tab of the wxKVG receiver interface. It includes sections for QuasiOptics (FlatMirror: OFF, Chopper: OFF), Cal (P-Cal: ON, NoiseSource: OFF), Rx Control (Pol: RHCP, Mode: SKY), and DownConverter Att Control (four channels with attenuation values like 5, 4, 3, 7). There are 'Update' and 'SetDefault' buttons at the bottom.

그림 5 wxKVG 수신기 창의 Control 탭

4. DBBC3 창

DBBC3의 제어와 감시도 wxDBBC3.py를 독립적으로 실행시키는 대신 wxKVG의 하위 창에서도 동일한 기능을 수행할 수 있다. 앞서 기술하였듯이 DBBC3 제어 창을 통해 DBBC를 구성하는 GGoMo 보드, DBBC 채널, Core3H보드의 제어와 상태 값을 읽어와 표시할 수 있다.

VLBI 관측 중에는 시각이 동기되어 있는지 VDIF 자료 출력이 되고 있는지 확인해야 한다. General 창과 마찬가지로 이들 주요 내용에 대해서는 상태에 따라 빨간색, 노란색, 또는 녹색으로 표시된다.

File

Sampling Mode
Mode: ddc, Sampling Rate (Msps): 32, Sampling Bit: 2
APPLY

DBBC DBBC IF
DBBC IF
IF: SYNTH, ATT: man, TARGET: 34000
A: off, B: off
Update

DBBC Control					DBBC Gain Control		DBBC Gain Status					
	FREQ	IF	BW	TPint	Mode/	GainL (0-255)	Mode GainU	GainL	TPUON	TPLON	TPUOFF	TPLOFF
CH1	640.00	A	16	1	SET	agc	122	116	15229	15107	0	0
CH2	672.00	A	16	1	SET	agc	115	101	15037	15254	0	0
CH3	704.00	A	16	1	SET	agc	104	100	15224	15104	0	0
CH4	736.00	A	16	1	SET	agc	108	105	15235	15180	0	0
CH5	768.00	A	16	1	SET	agc	114	117	15128	15100	0	0
CH6	800.00	A	16	1	SET	agc	136	148	14976	14875	0	0
CH7	832.00	A	16	1	SET	agc	144	156	14945	14900	0	0
CH8	864.00	A	16	1	SET	agc	166	160	15038	15027	0	0
CH9	640.00	B	16	1	SET	agc	132	127	15189	15252	0	0
CH10	672.00	B	16	1	SET	agc	120	122	15225	15213	0	0
CH11	704.00	B	16	1	SET	agc	10	11	99	92	0	0
CH12	736.00	B	16	1	SET	agc	133	140	15079	15152	0	0
CH13	768.00	B	16	1	SET	agc	147	161	15278	15081	0	0
CH14	800.00	B	16	1	SET	agc	171	194	14952	14794	0	0
CH15	832.00	B	16	1	SET	agc	192	213	14890	14979	0	0
CH16	864.00	B	16	1	SET	agc	227	216	14938	14843	0	0

Update Set Update Set

DCMD

```
>> core3h=1,sysstat
sysstat

System status:
Selected input : vsi1
Input sample rate : 128000000 Hz / 4
VSI input swapped : no
VSI input bitmask : 0xFFFFFFFF
VSI input width : 32 bit
PPS count : 94
TVG mode : vsi-h
```

그림 6 wxKVG의 DBBC3 창 - DBBC IF 설정 및 DBBC 채널 창

Sampling Mode
Mode: ddc_1
Sampling Rate (Mps): 32
Sampling Bit: 2
APPLY

DBBC DBBC IF
DBBC CONFIG
CHANNEL: 1:17:1
BAND WIDTH: APPLY
GAIN: APPLY
FREQ START: APPLY
FREQ STEP: APPLY

DBBC1-16 Core3H_1 Core3H_2
TIME / SYNC
Date Time: 2019-12-09T09:39:53
1PPS Count: 8155
syncd sys Sync

VDIFTX
stopped Start Stop
VSI Input
Input Select: vsi1
Sample Rate (sps): 128000000
Decimation: 2
SplitMode: off
VDIF Frame
BitWidth: 2
Num of Channels: 1
Payload Size: 8024
Corner Turning: off
Update Set

XG Destination
Index IP Address Thread
#1 0 10.10.1.13:46227 1 Set
#2 1 192.168.1.3:46227 1 Set
#3 2 192.168.1.4:46227 1 Set
#4 3 192.168.1.5:46227 1 Set

XG Arp Config
Dev Index MAC Address
#1 eth0 13 00:0A:CD:2C:FC:96 Set
#2 eth1 Set Set
#3 eth2 Set Set
#4 eth3 Set Set

XG Config
Dev IP Gateway Netmask
#1 eth0 10.10.1.11 10.10.1.1 28 Set
#2 eth1 193.0.0.0 193.0.0.0 28 Set
#3 eth2 171.0.0.0 171.0.0.0 28 Set
#4 eth3 171.0.0.0 171.0.0.0 28 Set

DCMD

그림 7 wxKVG의 DBBC 설정 및 Core3H 제어/감시 창

5. GSM 창

GSM은 GPU SpectroMeter(GSM)의 약자로 이 창을 이용해서 GSM의 설정을 변경할 수 있고 또한 현재 설정 상태를 확인할 수 있다. Data Acquisition 패널에서 Data Acquisition 체크 박스를 체크하면 GPU 스펙트로미터의 자료가 파일로 저장된다. 파일에 저장하는 모드는 테스트 목적으로 사용하기 위한 방식으로 일반 관측 중에는 파일에 저장하는 대신 TCP/IP의 지정된 port로 Client 프로그램의 접속이 있으면 자료를 출력하게 되어있다. DSLoc 프로그램에서 돌아가는 rcv_gsm.py는 이 data 출력 port에 접속하는 Client에 해당된다. IP Length로 지정된 시간 주기로 적분된 자료가 출력되면 출력되는 자료의 IF, 채널 개수 등을 GUI 창에서 제어할 수 있다. Update 버튼을 누르면 GPU 서버 프로그램에 설정된 값을 가져와 표시해준다.

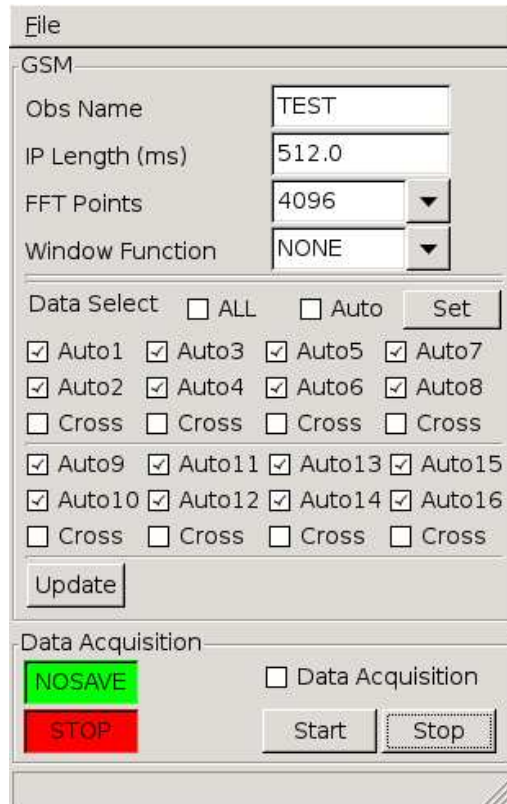


그림 8 GSM 상태 제어/표시 창

6. Mark6 상태 창

Mark6의 제어는 주로 스크립트에서 자동으로 제어가 이루어지고 wxKVG에서는 상태 표시 기능만 제공한다. 사용 중인 디스크모듈의 상태, 입력 스트림의 정보, 남아있는 디스크 공간, 마지막 기록 스캔의 정보가 표시된다. 이 중 기록 스캔에 대한 정보는 VLBI 관측 중에 스캔 기록이 끝났을 경우 갱신된다.

Mstat 패널은 디스크의 상태를 보여주는데 관측을 위해 열린 상태에는 노란색, 기록 중에는 녹색, 기록 명령이 전달되었지만 실제 기록기 안되는 pending 상태에는 주황색으로 표시된다. 디스크가 열리지 않은 상태에는 회색으로 표시된다.

File

MStat

Slot	Grp	eMSN	Size(GB)	Status	Protect	Type
1	1234	KVG00001/32000/4/8	31978	open	ready	sg
2	1234	KVG00002/32000/4/8	31978	open	ready	sg
3	1234	KVG00003/32000/4/8	31978	open	ready	sg
4	1234	KVG00004/32000/4/8	31978	open	ready	sg

Input Stream

Src	Format	Payload	Interface	IPAddress
		0/0		/0

Remaining HDD Space and Time

Time(min)	Space(GB)	Percent(%)	Rate(Mbps)
1065933.3	127912	99.9	16.0

Last Scan

Label	1234	6	344-0650_KV_TestRec2	vdif	15.3760
Status	OK	29.9150	2019y344d06h50m11s	4.11200	0

그림 9 wxKVG의 Mark6 상태 표시 창

7. Vex 제어 및 상태 창

Vex창은 Vex Control 탭과 Status 탭 TestVector 탭으로 구성된다. Vex Control 창은 VLBI 관측을 실행하는 스크립트와 vex 파일을 선택할 수 있다. 우주측지센터에서 DBBC3를 사용한 VLBI 관측은 runKVex_DBBC3.py 스크립트를 사용한다. vex 파일을 선택하여 Status 창에 표시된 내용을 확인 후 관측을 실행시킬 때는 아래 Execute 버튼을 누른다. SetUp 버튼은 관측에 필요한 수신기와 백엔드의 설정을 할 때 사용한다. Execute 시 실제 시스템 SetUp도 같이 진행된다. Abort 버튼은 진행 중인 관측을 종료시킬 때 사용한다.

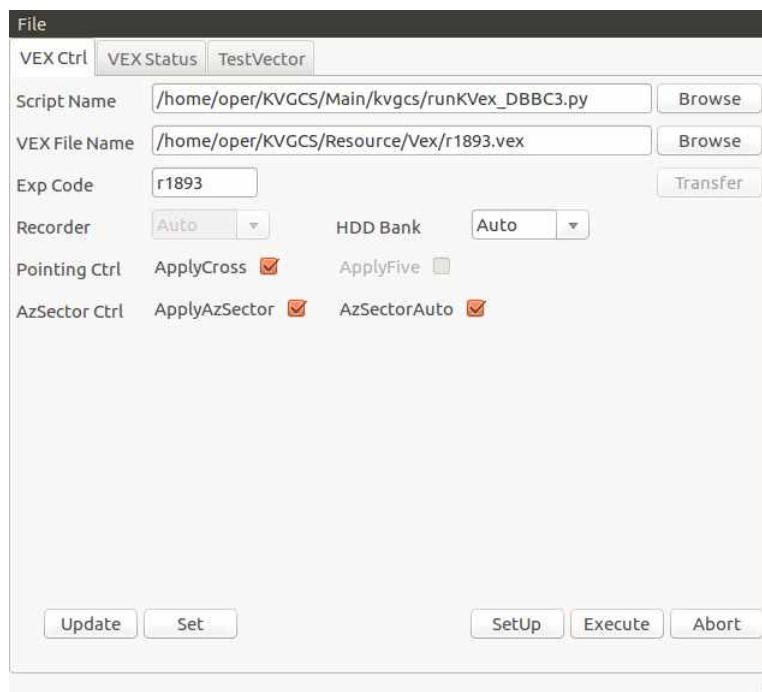


그림 10 wxKVG VLBI 관측 제어 탭

VEX Ctrl

VEX Status

TestVector

UT

2019-12-09 (343) 09:45:28

r19049k

2019 049 11:00:00 050 11:06:00

None

Recorder

Mark5B

1024 Mbps

Sync

Bank A

16000.0 GB

Mark6

0.0 Mbps

Sync

0 GB

3C286

S

	Source	Start	Dur	Stop	Mode	Pre:Mid:Post
2	0336-019	049 11:16:10	50	049 11:17:00	GEO1K_F	SDip::
3	0016+731	049 11:19:40	40	049 11:20:20	GEO1K	::
4	2353+816	049 11:21:00	120	049 11:23:00	GEO1K	::
5	0642+449	049 11:24:20	40	049 11:25:00	GEO1K	::
6	0945+408	049 11:26:30	100	049 11:28:10	GEO1K	::
7	0823+033	049 11:38:00	50	049 11:38:50	GEO1K	::
8	0748+126	049 11:39:30	40	049 11:40:10	GEO1K	::
9	0745+241	049 11:41:10	90	049 11:42:40	GEO1K	::
10	0834-201	049 11:44:00	110	049 11:45:50	GEO1K	::
11	0607-157	049 11:47:10	40	049 11:47:50	GEO1K	::
12	0528+134	049 11:49:10	80	049 11:50:30	GEO1K	::
13	0420-014	049 11:51:20	40	049 11:52:00	GEO1K	::

SETUP

START

ABORT

그림 11 wxKVG의 VLBI 관측 진행 상태 표시 탭

측지 VLBI 관측은 보통 24시간 동안 여러 천체를 바꿔가면서 빠른 시간 안에 하늘 여러 방향에 대한 지연량을 측정해야한다. 따라서 수십초의 길이의 여러 스캔으로 구성이 된다. 아래 그림은 r19049k 측지 관측의 Vex Status 창을 보여주는 데 현재 시스템 시각, 선택한 experiment의 관측 시간이 위에 표시된다. experiment code가 표시되는 창도 VLBI experiment가 실행 중이면 녹색으로 표시되면 동작 중이지 않을 때는 회색으로 표기된다. 관측 시간 범위 왼쪽 창은 스캔 사이에 천체를 바꾸면서 Antenna가 이동 중 일 때, 스캔 중 기록 중일 때에 따라 표시되는 문자와 색깔이 변경된다. 그리고 관측에 사용되는 기록기의 연결 상태와 기록 상태가 표시된다. 관측에 사용되는 기록기에 표시되는 색은 Mark6 창에서 디스크 모듈 상태와 같은 색으로 표시된다.

8. Obs Control 창

SDip, PS, Five 등과 같이 안테나 Calibration에 필요한 관측 모드에 대해 Total Power Detector와 Digital Spectrometer를 사용하여 측정하고 실시간 계산이 연동되어 있다. 아래 그림은 이런 관측 모드를 실행시키는 Obs Control 창을 보여준다. 각 탭에서 해당 관측 모드의 관측 파라미터를 설정할 수 있다.

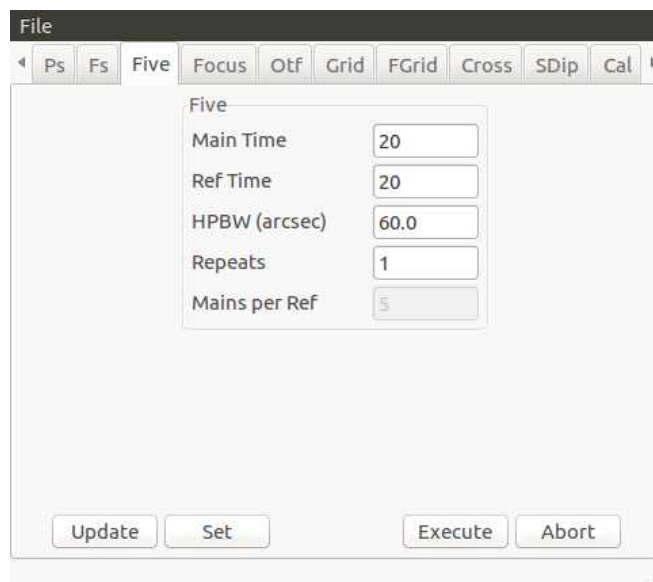


그림 12 관측 모드 제어 창 (Obs Control)

Obs Control 창에서 지원되는 각 관측 모드 탭에는 창 아래 부분에 Update, Set, Execute, Abort 버튼을 가지고 있다. 각 버튼의 기능은 다음과 같다.

- Update : 현재 KCS에 적용되어있는 파라미터 값을 가져와 표시한다.
- Set : 현재 창에 표시되어 있는 값들을 KCS에 적용한다. 혹은 개별 관측 파라미터 입력 칸에 값을 입력하고 Enter를 치면 파라미터가 KCS에 전달된다.
- Execute : 선택된 탭의 관측 모드를 실행시키는 명령을 KCS에 보낸다. 관측이 실행되면 KCS창에 관측 시작 메시지가 나타난다.
- Abort : KCS에서 진행 중인 관측을 중단시킨다. 관측이 중단되면 KCS 터미널에 관측이 강제 종료되었다는 메시지가 나타난다.

9. Obs Tool 창

Obs Tool 창은 관측 중 편의 제공을 위한 다양한 기능을 제공한다. ObsTool 창에서 주로 사용하는 4가지 탭을 차례로 살펴보면 다음과 같다.

가. Cata 탭

천체 목록 파일 선택 및 천체 목록 파일에서 천체 선택. Cata 탭에서 선택한 천체의 좌표는 Source 탭에 표시된다. 천체 목록 파일의 양식은 Gildas 패키지의 Astro 소프트웨어 양식을 따른다. 파일 확장자는 기본적으로 .SOU 확장자를 사용한다. 천체 목록이 표시된 창에서 천체를 클릭하면 창 하단에 현재 천체의 고도와 방위각과 은하 좌표계가 표시된다.

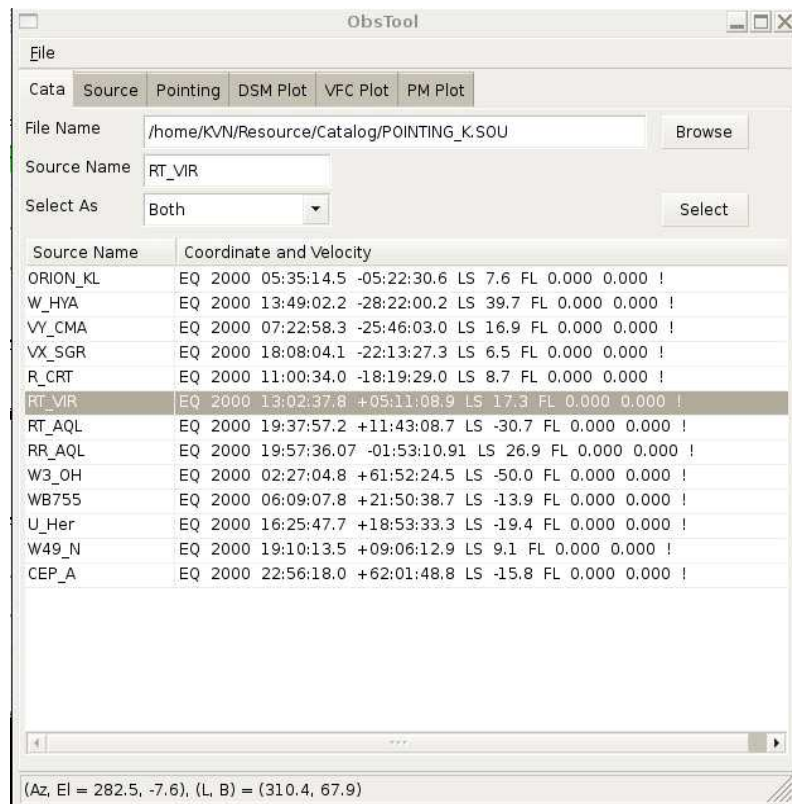


그림 13 ObsTool 창의 천체 목록 탭

나. Source 탭

안테나가 추적할 천체 좌표 입력. Set 버튼을 누르면 창에 표시된 좌표가 안테나에 전달된다. 안테나는 Main(천체) 과 Ref(비교위치) 두 개의 위치에 대한 정보를 가질 수 있어서 이 두 위치 사이의 교차 관측이 쉽게 구현되어있다. 창의 오른쪽 각 좌표계의 Off 값을 적용하면 천체와 비교 위치의 좌표에 상대적인 오프셋으로 줄 수도 있다.

Source Panel에서 행성을 선택하면 표시된 천체의 좌표와 상관없이 안테나는 행성의 위치를 계산해서 추적해나간다. 일반 천체를 추적하기 위해서는 Planet 창이 None으로 설정되어 있어야한다. Source Panel의 좌표계 (CoordSys) 창을 Ho(rizontal)로 선택하면 안테나는 천체 추적을 멈추고 현재 고도 방위각에 정지한 상태로 대기한다.

The image shows the 'Source' tab of the 'ObsTool' application. It contains several sections for setting coordinates and offsets for two positions: 'Main' and 'Ref'.

Source Section:

- Planet: None (dropdown)
- SourceName: "4C39.25" (text input)
- CoordSys: Eq (dropdown)
- Epoch: 2000 (text input)
- Vel (km/s): 0 (text input)
- VelSys: Lsr (dropdown)

Sky Section:

- Position: Main (dropdown)
- RaOffsetSys: Sky (dropdown)
- ZoneMode: Auto (dropdown)
- ZoneReq: 0 (text input)

Eq Section (Equatorial Coordinates):

	Ra	Dec
Main	09:27:03.01	39:02:20.9
Ref	09:27:03.01	39:02:20.9

Eq Off Section (Equatorial Offsets):

	Ra Off	Dec Off
Main	00:00:00.00	00:00:00.0
Ref	00:00:00.00	00:00:00.0

Ga Section (Galactic Coordinates):

	L	B
Main	183.70853676	46.163715157
Ref	183.70853676	46.163715157

Ga Off Section (Galactic Offsets):

	L Off	B Off
Main	0.0	0.0
Ref	0.0	0.0

Ho Section (Horizontal Coordinates):

	Az	El
Main	74.1707251455	57.0964874345
Ref	74.1707251455	57.0964874345

Ho Off Section (Horizontal Offsets):

	Az Off	El Off
Main	0.0	0.0
Ref	0.0	0.0

Buttons: Set, Update

그림 14 ObsTool의 천체 좌표 제어 창

다. Pointing 탭

Pointing 탭은 안테나의 지향 오프셋 조정과 초점 조정에 사용된다. 현재 적용되어있는 값을 같이 표시한다. 천체 추적을 위한 좌표 값 전송 주기도 이 탭에서 조절할 수 있다.

그림 15 안테나 지향 오프셋 및 초점 조절 창

라. DSM Plot 탭

DSM Plot 탭은 분광자료의 Plot과 처리에 필요한 파라미터를 입력하기 위해 사용한다. 가장 위의 Scan 패널에서는 최종 캘리브레이션 된 단일경 측정 자료를 저장할 CLASS 양식의 파일 이름을 선택하는데 사용한다.

Line Window 패널은 분광 자료의 baseline 제거 및 선 면적 적분을 위한 파라미터를 입력하기 위해 사용한다. Five 또는 Focus 관측 전 방출선의 채널 위치를 확인해서 이 파라미터들을 설정해야 의미 있는 처리 결과를 얻을 수 있다.

BaseDegree는 baseline 제거 시 다항함수의 차수를 나타낸다. BaseDegree가 -1 값이면 baseline 제거를 하지 않는데 연속과 천체를 이용한 측정에서는 -1을 선택해야한다.

Plot Panel은 하나의 스캔 관측 후 표시되는 스펙트럼 창에서 채널과 안테나 온도의 범위를 지정하는데 사용한다. Quick Look Stream은 분광기 제어 소프트웨어에서 DSLoc로 전달되는 스펙트럼 자료를 실시간으로 Plot을 보여 줄 스트림을 선택하는데 사용한다. 이 창에서는 하나의 스트림만 실시간 Plot으로 선택 할 수 있다.

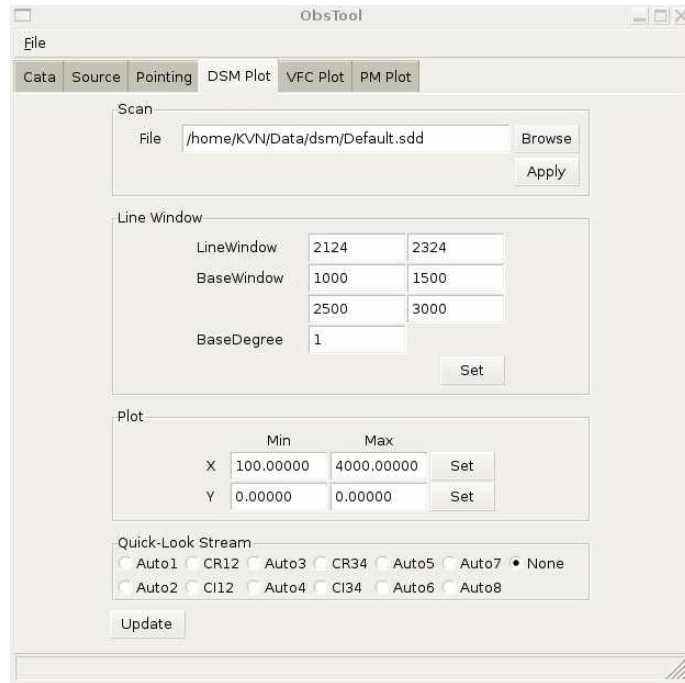


그림 16 ObsTool 창 의 DSM Plot 탭

10. 관측 환경

아래 그림과 같이 전체 관측 환경은 GUI와 터미널, Plot가 같이 표시된다. GUI는 관측 또는 테스트 목적에 따라 필요한 창을 열어서 사용할 수 있도록 구성되어있다.

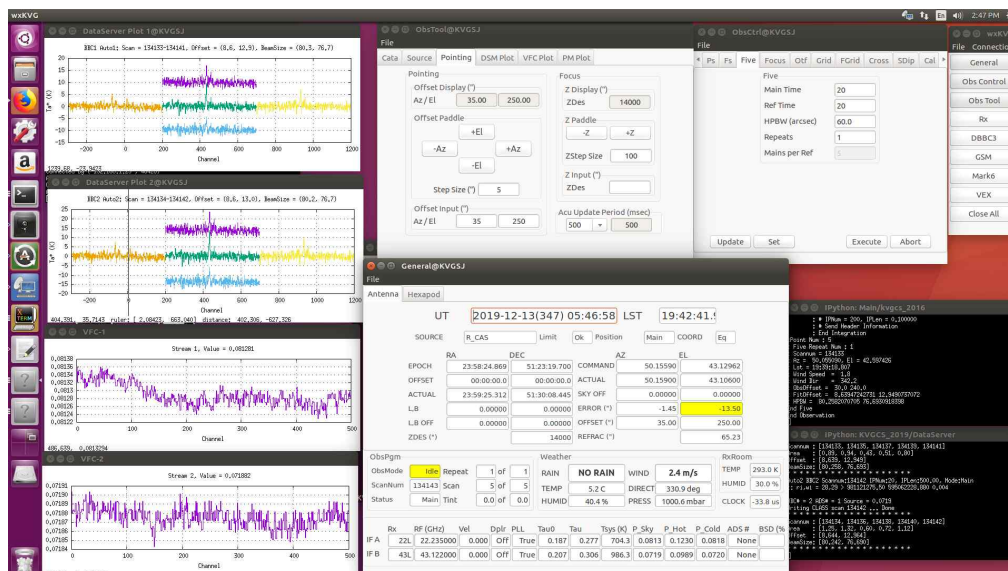


그림 17 KVGCS Five 관측 화면

III. VLBI 관측 매뉴얼

1 VLBI 관측 일정 파일 확인

VLBI 관측은 멀리 떨어져 있는 독립된 여러 망원경들이 동시에 같은 천체를 바라보도록 망원경을 제어하고 약속한 주파수와 자료 양식으로 수신되는 신호를 저장매체에 기록해야한다. 이런 목적으로 약속된 관측을 위한 시스템 설정과 시간에 따른 관측 천체, 기록 시간 등의 정보를 VLBI 관측 스케줄 파일에 기록하여 이 스케줄 파일을 이용하여 관측을 수행한다. 세계적으로 널리 공통으로 사용하고 있는 관측 스케줄 파일은 skd 양식과 vex 양식 두 가지 양식을 사용하고 있는데 두 가지 양식은 서로 호환이 가능하다. 즉 skd 양식으로 정의된 관측은 측지 VLBI 스케줄 소프트웨어인 sked를 이용하여 vex 파일로 변환할 수 있다. KVGCS 에서는 이러한 sked로 생성한 vex 파일을 수정 없이 그대로 해석해서 사용할 수 있다.

check_vex.py 프로그램은 vex 파일을 해석해서 결과를 출력해주는 프로그램이다. 이 프로그램은 여러 가지 option을 제공하는데 아래와 같이 -h option으로 실행시키면 사용 가능한 option을 보여준다.

```
oper@KVG:~/KVGCS/bin$ check_vex.py -h
./check_vex.py -s station name -p proposal season <-f vex filename> <-e
extension> <-v> <-q> <-a> obs_code1 <obs_code2> ... <obs_codeN>
-s: station name, default KVGSI
-p: proposal season
-e: extension of vex file, default .vex
-f: vex file
-v: verbose
-a: allow past
-q: make queue
-h: help
```

다음은 check_vex.py 를 이용하여 출력한 결과이다. -v 옵션으로 자세한 내용을 출력하였다.

```
oper@KVG:~/KVGCS/bin$ check_vex.py -a -v r1893
##### r1893 #####
# From SCAN 1 127-1700a
Bandwidth : 8.0
NChannel : 16
Sample Bit : 2
Data Rate : 512.0
# bbc_tbl :
{'&BBC01': (1, 'X1R', 1, '&IF_1N', 7680.0),
```

```

'&BBC02': (2, 'X1R', 1, '&IF_1N', 7680.0),
'&BBC03': (3, 'X1R', 1, '&IF_1N', 7680.0),
'&BBC04': (4, 'X1R', 1, '&IF_1N', 7680.0),
'&BBC05': (5, 'X2R', 1, '&IF_3N', 8080.0),
'&BBC06': (6, 'X2R', 1, '&IF_3N', 8080.0),
'&BBC07': (7, 'X2R', 1, '&IF_3N', 8080.0),
'&BBC08': (8, 'X2R', 1, '&IF_3N', 8080.0),
'&BBC09': (9, 'SR', 1, '&IF_2N', 1600.0),
'&BBC10': (10, 'SR', 1, '&IF_2N', 1600.0),
'&BBC11': (11, 'SR', 1, '&IF_2N', 1600.0),
'&BBC12': (12, 'SR', 1, '&IF_2N', 1600.0),
'&BBC13': (13, 'SR', 1, '&IF_2N', 1600.0),
'&BBC14': (14, 'SR', 1, '&IF_2N', 1600.0)}

```

```

# chan_tbl :
{'&BBC01': ['&CH01', '&CH02'],
 '&BBC02': ['&CH03'],
 '&BBC03': ['&CH04'],
 '&BBC04': ['&CH05'],
 '&BBC05': ['&CH06'],
 '&BBC06': ['&CH07'],
 '&BBC07': ['&CH08'],
 '&BBC08': ['&CH09', '&CH10'],
 '&BBC09': ['&CH11'],
 '&BBC10': ['&CH12'],
 '&BBC11': ['&CH13'],
 '&BBC12': ['&CH14'],
 '&BBC13': ['&CH15'],
 '&BBC14': ['&CH16']}

```

```

# channel_assign:
{'&BBC01': ('X1R', [1, 2, 3, 4, 5]),
 '&BBC05': ('X2R', [6, 7, 8, 9, 10]),
 '&BBC09': ('SR', [11, 12, 13, 14, 15, 16])}

```

```

# IF Selector Setting
Input: ['X1R', 'X2R', 'SR']
Number: [2, 3, 1]

```

```

# DBBC setting parameters
DBBC01: {'freq': 616.0, 'bw': 8, 'tpint': 1, 'channel': 'A'}
DBBC02: {'freq': 656.0, 'bw': 8, 'tpint': 1, 'channel': 'A'}
DBBC03: {'freq': 756.0, 'bw': 8, 'tpint': 1, 'channel': 'A'}
DBBC04: {'freq': 916.0, 'bw': 8, 'tpint': 1, 'channel': 'A'}
DBBC05: {'freq': 672.0, 'bw': 8, 'tpint': 1, 'channel': 'B'}
DBBC06: {'freq': 792.0, 'bw': 8, 'tpint': 1, 'channel': 'B'}
DBBC07: {'freq': 852.0, 'bw': 8, 'tpint': 1, 'channel': 'B'}
DBBC08: {'freq': 872.0, 'bw': 8, 'tpint': 1, 'channel': 'B'}
DBBC09: {'freq': 688.0, 'bw': 8, 'tpint': 1, 'channel': 'C'}
DBBC10: {'freq': 708.0, 'bw': 8, 'tpint': 1, 'channel': 'C'}
DBBC11: {'freq': 728.0, 'bw': 8, 'tpint': 1, 'channel': 'C'}

```

```
DBBC12: {'freq': 758.0, 'bw': 8, 'tpint': 1, 'channel': 'C'}
DBBC13: {'freq': 808.0, 'bw': 8, 'tpint': 1, 'channel': 'C'}
DBBC14: {'freq': 828.0, 'bw': 8, 'tpint': 1, 'channel': 'C'}
DBBC15: {'freq': 0, 'bw': 8, 'tpint': 1, 'channel': 'C'}
DBBC16: {'freq': 0, 'bw': 8, 'tpint': 1, 'channel': 'C'}
```

```
# Core3H_1 Setting:
```

```
# VDIF Frame
```

```
VSI Input      : vsi1
Sample Rate    : 128000000
Decimation Rate : 8
Split mode     : off
```

```
# VDIF Frame
```

```
Sample Bit     : 2
Num of Channels : 5
Payload Size   : 8192
```

```
# Core3H_2 Setting:
```

```
# VDIF Frame
```

```
VSI Input      : vsi1
Sample Rate    : 128000000
Decimation Rate : 8
Split mode     : off
```

```
# VDIF Frame
```

```
Sample Bit     : 2
Num of Channels : 5
Payload Size   : 8192
```

```
# Core3H_3 Setting:
```

```
# VDIF Frame
```

```
VSI Input      : vsi1
Sample Rate    : 128000000
Decimation Rate : 8
Split mode     : off
```

```
# VDIF Frame
```

```
Sample Bit     : 2
Num of Channels : 6
Payload Size   : 8192
```

```
# Frequency Parameters
```

```
BBC1 {'Input': 'X1R', 'RF': 8.296, 'SideBand': 'USB', 'LO': 7.68, 'IFc': 0.0}
BBC2 {'Input': 'X2R', 'RF': 8.752, 'SideBand': 'USB', 'LO': 8.08, 'IFc': 0.0}
BBC3 {'Input': 'SR', 'RF': 2.2880000000000003, 'SideBand': 'USB', 'LO': 1.6, 'IFc': 0.0}
```

```
# Recorder Information
```

```

Recorders      : 'Mark6'
Recording Rate : 512Mbps
Total recording time (sec) : 55039

```

```

# Target Sources
Source catalog file : /home/oper/KVGCS/Resource/Catalog/EXPER/r1893.sou

```

```

#####
# Experiment Summary
ObsCode      =          'r1893'
ObsType      =          'vlbi'
Session      =          'test'
PI_Name      =          'NASA'
Title        =          'IVS-R1'
StartTime    = '2019-05-07_17:00:00'
StopTime     = '2019-05-08_16:59:55'
Frequency    =          'X1R,X2R,SR'
Mode         =          '8x16MHz'
Recorder     =          'Mark6'
DataSize     =          3522.496
RegistrationTime =          ""
DataRate     =          512
Status       =          'PAST'

```

-v option을 부여한 경우 vex 파일을 해석해서 IF Selector, Receiver와 DBBC3의 설정 파라미터를 모두 추출해서 출력한다. 관측 운영에 필요한 내용은 Experiment Summary 부분에 별도로 출력하는데 이 내용은 여러 관측을 자동 실행시키기 위한 관측 큐의 정보로 저장된다. -q option을 주면 관측 요약 정보를 담은 큐 파일이 생성 된다. 큐 파일이 생성되는 디렉토리나 파일 이름 양식은 다음과 같다.

```

큐 파일 디렉토리: $KCS_RESOURCE/KQueue/
파일 이름 양식 : <experiment start time>_<obs_code>.kqueue

```

KQueue 개체는 큐 파일을 생성하는 regist() 함수를 지원하는데 기본 설정은 관측 종료 시간이 생성 시간보다 늦은 경우 큐 파일을 생성을 하지 않는다. 테스트 목적으로 과거 관측의 큐 파일을 만들기 위해서는 check_vex.py 를 실행시킬 때 -a 옵션을 준다.

2. Vex GUI 창을 이용한 관측 선택 및 실행

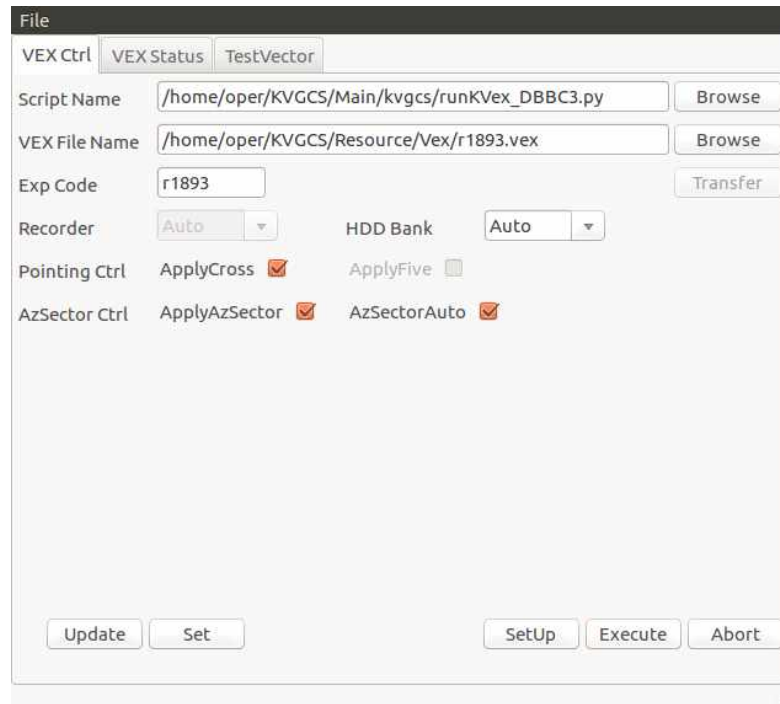


그림 1 Vex Control 창

wxGUI의 Vex Control 창에서도 vex 파일을 선택하면 내용을 해석해서 Vex Status 창에 관측 코드, 관측 시작과 종료 시간, 사용하는 기록기가 헤더 부분에 표시되고 아래에 각 스캔의 천체와 기록 시작 시각, 기록 시간, 모드, 캘리브레이션 작업 정보를 보여준다.

관측 시작 전 이미 안테나의 지향 정밀도와 초점 등을 확인을 했다면 바로 SetUp 버튼을 눌러 vex 파일에서 기술된 관측을 수행하기 위한 수신기와 백엔드 설정을 할 수 있다. 안테나 지향 정밀도와 초점 등의 정상적인지 확인이 안되었을 경우는 앞의 1장에 기술된 과정에 따라 시스템의 상태를 확인한다. 기본적인 시스템 이득 최적화와 캘리브레이션 (AGC, SDip) 과정은 runKVex_DBBC3.py 스크립트에서 자동으로 수행한다. 혹은 vex 파일에서 첫 스캔 앞에 이런 과정을 수행하도록 설정해놓을 수 있다.

Setup 버튼은 시스템 설정만 수행한다. VLBI 관측에 적절하게 설정이 되어있는지 확인할 때 사용할 수 있다. Setup 설정 시에는 안테나는 움직이지 않지만 그 외의 수신기부터 기록기까지 관측에 필요한 설정이 적용된다. 각 GUI 창을 통해 필요한 설정이 정상적으로 적용되었는지 확인한다.

Execute 버튼은 시스템 설정 뿐 아니라 실제 관측 스캔에 맞춰서 VLBI 관측을 진행한

다. 이를 위해 망원경과 기록기를 지정된 스캔 일정에 맞춰서 제어한다. Execute 버튼으로 관측 실제 시작 시 시스템 설정이 명령이 다시 적용된다. 이 때는 안테나도 천체를 향해 움직인다. 안테나가 천체에 도달하면 캘리브레이션을 수행 후 첫 스캔의 기록 시작 시간까지 기다렸다가 기록기에 기록 명령을 보낸다. 기록 시작 명령에 기록기가 정상적으로 기록을 시작하면 Mark6 표시 색깔이 녹색으로 변하는지 확인한다.

The screenshot shows the 'VEX Status' window. At the top, there are tabs for 'VEX Ctrl', 'VEX Status', and 'TestVector'. Below the tabs, the 'UT' section displays the date and time '2019-12-09 (343) 09:45:28'. The 'Recorder' section shows 'r19049k' and '2019 049 11:00:00 050 11:06:00'. Below this, there are buttons for 'Mark5B' and 'Mark6' (highlighted in yellow), along with '1024 Mbps' and '0.0 Mbps' labels. There are also 'Sync' and 'Bank A' buttons, and a '16000.0 GB' label. At the bottom, there is a '3C286' label and a 'S' button. A table with 7 columns (Source, Start, Dur, Stop, Mode, Pre:Mid:Post) lists 13 scans. The table is followed by 'SETUP', 'START', and 'ABORT' buttons.

	Source	Start	Dur	Stop	Mode	Pre:Mid:Post
2	0336-019	049 11:16:10	50	049 11:17:00	GEO1K_F	SDip::
3	0016+731	049 11:19:40	40	049 11:20:20	GEO1K	::
4	2353+816	049 11:21:00	120	049 11:23:00	GEO1K	::
5	0642+449	049 11:24:20	40	049 11:25:00	GEO1K	::
6	0945+408	049 11:26:30	100	049 11:28:10	GEO1K	::
7	0823+033	049 11:38:00	50	049 11:38:50	GEO1K	::
8	0748+126	049 11:39:30	40	049 11:40:10	GEO1K	::
9	0745+241	049 11:41:10	90	049 11:42:40	GEO1K	::
10	0834-201	049 11:44:00	110	049 11:45:50	GEO1K	::
11	0607-157	049 11:47:10	40	049 11:47:50	GEO1K	::
12	0528+134	049 11:49:10	80	049 11:50:30	GEO1K	::
13	0420-014	049 11:51:20	40	049 11:52:00	GEO1K	::

그림 2 wxKVG의 VLBI 관측 진행 상태 표시 탭

3. 관측 실행 전/ 진행 중 확인 내용

안테나, 수신기, DBBC3, Mark6에서 관측 종류에 따라 주의해서 확인해야하는 부분을 정리하면 다음과 같다.

가. 안테나

- 지향 오프셋, 초점 값 (ZDes) 확인
- ZDes 값이 $2/8\text{GHz} = 0$, $22/43\text{GHz} = 14000$

나. 시각 시스템 및 기상 관측 시스템

- 시각 오프셋 (General 창에 표시)
- 기상 값 (General 창 표시)이 갱신되고 있는지 확인 (혹은 터미널에서 tailw로 확인)

다. 수신기

- 2/8 또는 22/43 관측에 따른 FlatMirror 상태 (수신기 상태 창)
- 22/43GHz 관측의 경우 Chopper 상태 (수신기 상태 창)
- 2/8GHz 관측의 경우 입력 모드가 Sky 상태 (수신기 상태 창)
- SDip 실행 후 표시되는 시스템 잡음 온도 (General 창)
- Total Power Detector Plot에서 표시되는 신호 세기 (~ 0.1 근처)
- 주파수 Lock 상태 (General 창 하단 Rx, PLL 상태)
- 편파기 선택 상태 (General 창 하단 Input에 표시)

라. DBBC3

- 시각 동기화와 VDIF 스트림 출력 상태 (DBBC3 Core3H 탭)
- Data Rate 및 시각 오프셋 (Mark6 또는 GSM에서 vdfitimeUDP로 확인)

마. Mark6

- 디스크의 open 상태 및 남은 공간
- 기록 중 디스크 잔여 공간의 변화
- 스캔 기록 후 기록 스캔의 체크 결과
- DBBC3로부터 입력되는 VDIF 시각의 시각차 (vdfitimeUDP로 확인)
Mark6-Sj1\$ vdfitimeUDP -cpu=2 -offset=8 <port number>
or
\$ checkvdif1 < or checkvdif2 >
- 입력 스트림으로부터 스펙트럼 체크
Mark6-SJ1:~/test/\$ dbbc_ql.sh <data_rate> <# of channels> <port number>
- 입력 스트림을 처리하는 ring buffer 상태 확인
Mark6-SJ1:\$ dpstat Mark6-SJ1

4. 관측 실행 중 로그 및 메시지 확인

가. Station Log

관측 중 진행 상황에 대해 실시간으로 Station 로그가 생성된다. (Station Log 내용에 대한 자세한 내용은 부록을 참고하라.)

디렉토리 : /home/kvg/KVGCS/Log/VLBI
파일명 : <obs_code><Site Code>.log
링크명 : KVGCS.log

- Station Log의 기록 상태 확인

oper@KVG:~\$ tailw

```

STATION_LOG@KVGSI
050105949/ TPREC_OFF/
050105950/ MK6_SCAN/ 0, 2, 3480, r19049k_KVGSI_050105909, 1, -, OK, vdif, 2019y
050d10h59m10s, 39.426, 10.132, 2.056, 0
# SCAN Start, Stop Time : 2019-02-19 11:00:50, 2019-02-19 11:02:50
# No00506, GEO1K
# Auto1,2,100,0,50,0,USB,1;Auto2,1,100,0,50,0,USB,2
050105950/ SOURCE/ 2353+816, $, 23, 56, 22.793799, 81, 52, 52.25521, 2000,0, 0.
0, 0,0
050110007/ ONSOURCE/
050110007/ WX/ 0,4, 997.9, 99,0
050110050/ MK6_REC/ ON
050110050/ ST/
050110050/ TPREC_ON/ 134035
050110057/ TSYS/ 01, 37.1733032218, , 123.755491123
050110057/ TSYS/ 02, 37.1733032218, , 233.177667695
050110057/ DSM_BSD/
050110057/ PTOFF/ 0,0, 270,0
050110249/ MK6_REC/ OFF
050110249/ ET/
050110249/ TPREC_OFF/
050110250/ MK6_SCAN/ 0, 2, 3481, r19049k_KVGSI_050110049, 1, -, OK, vdif, 2019y
050d11h00m50s, 119.468, 30.696, 2.056, 0
# SCAN Start, Stop Time : 2019-02-19 11:04:10, 2019-02-19 11:04:50
# No00507, GEO1K_L
# Auto1,2,100,0,50,0,USB,1;Auto2,1,100,0,50,0,USB,2
050110250/ SOURCE/ 0642+449, $, 06, 46, 32.025985, 44, 51, 16.59013, 2000,0, 0.
0, 0,0
050110317/ ONSOURCE/
050110317/ WX/ 0,4, 997.9, 99,0
050110410/ MK6_REC/ ON
050110410/ ST/
050110410/ TPREC_ON/ 134041
050110417/ TSYS/ 01, 73.0779314841, , 110.747914053
050110417/ TSYS/ 02, 73.0779314841, , 221.138385903
050110417/ DSM_BSD/
050110417/ PTOFF/ 0,0, 270,0
050110449/ MK6_REC/ OFF
050110449/ ET/
050110449/ TPREC_OFF/
050110450/ MK6_SCAN/ 0, 2, 3482, r19049k_KVGSI_050110409, 1, -, OK, vdif, 2019y
050d11h04m10s, 39.39, 10.122, 2.056, 0
050110805/ SECZ/ 01, 0.0745812429861, 0, 263.6, 0,0, 0,0
050110805/ SECZ/ 02, 0.07300494376, 0, 263.6, 0,0, 0,0
050110821/ CLOCK/ -30,5
050110821/ MK6_RTINE/ 0, 2, 2048,0, 94792.96875, 24267, 32000
# Transferring log file to AOC
# End File Transfer
# End Experiment : r19049k

```

그림 3 Station Log 예

- 관측 진행 중 Station Log에서 기록 스캔 정보 (MK6_SCAN), 시스템 온도 (TSYS) 확인
- 관측 종료 후 Station Log를 지정 서버로 자동 전송 확인

나. 이메일 메시지

관측 진행 내용과 에러 발생시 gmail로 지정한 계정에 자동으로 메일을 발송한다. 이와

관련된 설정 방법은 부록에 기술하였다.

- Setup 버튼을 누른 경우 관측 Observation Ready 메일 전송 확인
- 첫 스캔 기록 시작 시 Observation Start 메일 전송 확인
- 관측 종료 후 Observation Start 메일 전송 확인
- 관측 중 에러 메시지에 대한 메일 수신 여부 확인

5. 2개 이상의 연속 관측

일반적으로 VLBI 관측은 관측 스케줄 단위로 관측을 수행하도록 되어있고 K/Q 관측 소프트웨어도 개별 스케줄 파일을 선택해서 관측을 진행하도록 되어있다. 관측의 편의를 위해 두 개 이상의 관측 스케줄 파일을 선택하여 스케줄 파일에 기록되어 있는 관측 시간 순서대로 자동으로 순차적으로 관측을 수행하는 기능을 구현한다. 각 관측 시작 시에 필요한 시스템 설정 및 필요한 Calibration을 자동으로 수행한다.

vex 파일을 읽어서 큐를 생성하는 기능을 check_vex.py에 구현하였고 -q option을 주면 vex 파일을 체크해서 KQueue 디렉토리에 kqueue 파일을 생성한다. 이 kqueue 파일을 관측 시작 시간, 관측 명, PI, 주파수 등 관측 일정 관리에 필요한 주요 정보를 포함하고 있다. 큐의 디렉토리 위치와 파일 이름, 큐에 기술된 내용은 다음과 같다.

- 큐 파일 디렉토리: \$KCS_RESOURCE/KQueue/
- 파일 이름 양식 : <experiment start time>_<obs_code>.kqueue
- 큐 파일 예: 2019-05-07_17:00:00_r1893.kqueue

```
ObsCode      = 'r1893'
ObsType      = 'vlbi'
Session      = 'test'
PI_Name      = 'NASA'
Title        = 'IVS-R1'
StartTime    = '2019-05-07_17:00:00'
StopTime     = '2019-05-08_16:59:55'
Frequency    = 'X1R,X2R,SR'
Mode         = '8x16MHz'
Recorder     = 'Mark6'
DataSize     = 3522.496
RegistrationTime = ''
DataRate     = 512
Status       = 'PAST'
```

QueueManager 스크립트 kqueue_manager.py는 주 관측 프로그램 KVCS의 Thread 중 Scheduler Console Thread의 내부 subthread로 돌아간다. QueueManager를 동작시키는

명령은 다음과 같다.

```
KVGCS> sch_con.openMainLoop('kqueue_manager.py')    # 실행
KVGCS> sch_con.isAlive()                             # 실행 상태 확인
KVGCS> sch_con.stop()                                # 실행 중단
```

QueueManager는 시스템 설정과 Calibration을 위한 시간을 고려해서 KQueue 파일에 기록된 시작 시간보다 10분 앞서 관측을 시작 시킨다. 이 시간 간격은 kqueue_manager.py에 정의되어있는 PRESTART_TIME 값으로 쉽게 조정할 수 있다.

IV. 안테나 캘리브레이션 라이브러리 매뉴얼

1. 라이브러리 구조

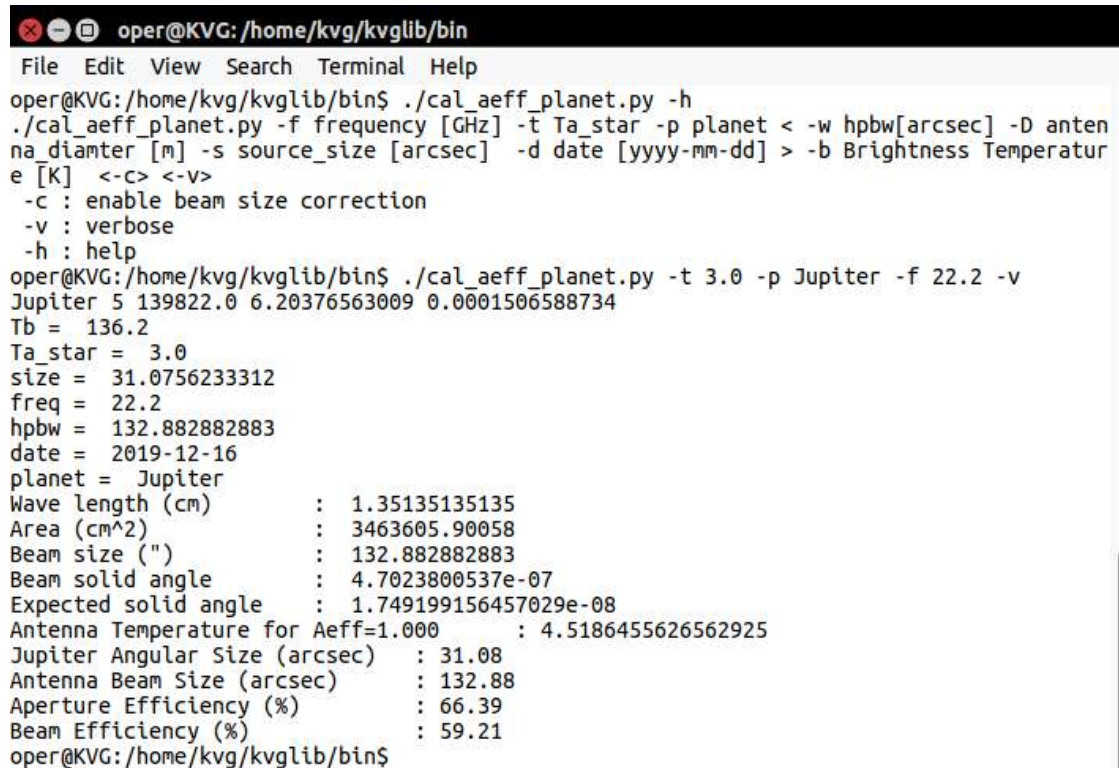
KVGCS에서 사용하는 안테나 성능 측정에 사용하는 모듈들을 별도의 라이브러리로 묶어서 독립적으로 사용할 수 있도록 구성하였다. 라이브러리의 명칭은 kvglib로 명칭을 정했으며 라이브러리의 구조는 다음과 같다.

표 1 전파망원경 Calibration 라이브러리 구조

모듈 / 파일 경로명			기능
kvglib	/utli	/coord	천체 좌표 변환
		/calibration	안테나 / 수신기 Calibration
		/unitconv	단위 변환
		/analysis	fitting / 해석
	/bin	/cal_aeff_planet.py	안테나 온도로부터 행성을 이용한 효율 계산
		/caleff_x.py	안테나 온도로부터 Crab을 이용한 효율 계산
		/cal_sdip.py	고도에 따른 파워 측정값으로 대기 투과율 계산 & Plot
		/noise_sp.py	Noise Spectrum 분석 Plot
		/adev_pl.py	이득 안정도 분석 Plot

위 라이브러리의 모듈의 사용을 위해 numpy, matplotlib, scipy, ipython 패키지가 설치되어야 한다.

2. 안테나 구경 효율 계산 프로그램



```
oper@KVG: /home/kvg/kvglib/bin
File Edit View Search Terminal Help
oper@KVG:/home/kvg/kvglib/bin$ ./cal_aeff_planet.py -h
./cal_aeff_planet.py -f frequency [GHz] -t Ta_star -p planet < -w hpbw[arcsec] -D antenna_diameter [m] -s source_size [arcsec] -d date [yyyy-mm-dd] > -b Brightness Temperature [K] <-c> <-v>
-c : enable beam size correction
-v : verbose
-h : help
oper@KVG:/home/kvg/kvglib/bin$ ./cal_aeff_planet.py -t 3.0 -p Jupiter -f 22.2 -v
Jupiter 5 139822.0 6.20376563009 0.0001506588734
Tb = 136.2
Ta_star = 3.0
size = 31.0756233312
freq = 22.2
hpbw = 132.882882883
date = 2019-12-16
planet = Jupiter
Wave length (cm) : 1.35135135135
Area (cm^2) : 3463605.90058
Beam size (") : 132.882882883
Beam solid angle : 4.7023800537e-07
Expected solid angle : 1.749199156457029e-08
Antenna Temperature for Aeff=1.000 : 4.5186455626562925
Jupiter Angular Size (arcsec) : 31.08
Antenna Beam Size (arcsec) : 132.88
Aperture Efficiency (%) : 66.39
Beam Efficiency (%) : 59.21
oper@KVG:/home/kvg/kvglib/bin$
```

그림 1 행성을 이용한 효율 계산 프로그램 실행 예

안테나 구경 효율 계산 프로그램인 cal_aeff_planet.py는 측정된 행성의 안테나 온도로 부터 안테나의 효율을 계산해주는 프로그램이다. 정확한 계산을 위해서는 행성의 밝기 온도와 크기가 필요하다. 행성의 크기는 천체 좌표 계산 모듈을 통해 계산한다. 행성의 밝기 온도는 기존에 알려진 행성의 밝기 온도를 테이블로 사용한다.

표 2 행성의 밝기 온도

	밝기 온도 (K)				
	22GH z	43GH z	86GH z	100G Hz	130G Hz
금성	505±25 ^b	450±32 ^d	367 ^{e,g}	358 ^e	340 ^e
목성	136.2 ^a	154.8 ^a	173.5 ^a	172	172.0 ^a
화성		195 ^c	200 ^c	193 ^c	203 ^c
토성	123	133±13 ^d	149±4 ^f		
천왕 성	166	153	120		107
해왕 성	161	126	142		125

위 표에서 제시된 행성의 밝기 온도의 출처는 다음과 같다.

a: Weiland et al. 2011 ApJS

b: Butler, B. J., et al. 2001, Icarus, 154, 226

c: www.aoc.nrao.edu/~bbutler/work/mars/model

www.lesia.obspm.fr/perso/emmanuel-lellouch/mars

d: Greve, A. et al. 1994, A&A 286, 654

e: Fahd, A. K., 1992, NASA Technical Report No. 1992-1

f: Mangum, J., 2000, User's Manual for the NRAO 12 Meter Millimeter-Wave Telescope, 116

g: Ulich, B. L., 1981, Millimeter-wavelength Continuum Calibration Sources, AJ, 86, 1619

행성의 밝기 온도는 행성에 따라 수 K 정도의 오차가 있을 수 있는데 목성의 밝기 온도가 비교적 정확하게 잘 알려져 있다. 금성의 밝기 온도는 태양과의 위치에 따라 차이가 있을 수 있고 15-20K 정도의 오차가 있다. 또한 금성은 각 크기가 지구와의 거리에 따라 10초에서 50초 정도 까지 크게 변할 수 있기 때문에 망원경의 빔 크기에 비해 큰 경우는 측정 오차가 생길 수 있는 것을 고려해야한다. 화성의 밝기 온도는 시간에 따라 조금씩 변하는데 두 웹페이지에서 계산할 수 있다. 토성의 경우 토성의 고리의 기울어진 각도에 따라 밝기 온도가 차이가 생길 수 있기 때문에 상대적으로 큰 오차를 줄 수 있다.

따로 행성의 정확한 온도를 알고 있다면 프로그램에 저장된 값을 사용하는 대신 -b 옵션으로 밝기 온도를 입력할 수 있다. 안테나의 빔 크기를 아는 경우 -w 옵션으로 빔 크기를 입력할 수 있다. 만약 행성의 Cross Scan 관측을 통해 행성의 크기와 안테나 빔이

Convolution된 빔 크기를 아는 경우 -w 옵션으로 측정된 Convolved 빔 크기를 주고 -c 옵션을 주면 Convolution 된 효과를 보정한 빔 크기를 계산해서 효율 계산에 사용한다. 빔 크기에 대한 아무 옵션도 주지 않는 경우 빔 크기를 파장과 안테나 직경의 비 ($= \lambda/D$) 를 이용해서 계산한 값을 사용한다.

위 계산을 위해 cal_aeff_planet.py는 util.coord 모듈의 planet_size() 함수와 util.calibration 모듈의 calTa() 함수를 사용한다.

3. 대기 투과율 계산 프로그램

cal_sdip.py 프로그램은 고도에 따라 대기의 파워를 측정했을 때 이 값을 이용하여 대기의 투과율을 계산하는 프로그램이다. 이 프로그램을 실행하기 위해서는 고도에 따른 파워의 크기를 측정한 자료 파일이 필요하다. 아래 그림에서와 같이 sdip_test.dat 파일은 고도와 파워의 2열을 갖는 파일인데 고도 0도에서의 파워는 수신기를 상온 흑체로 가렸을 경우의 파워를 나타낸다.

대기 투과율 계산 모듈은 util.calibration.calSDip 모듈을 이용하는데 이 모듈은 Airmass ($= 1/\sin(EL)$) 에 따른 대기의 파워를 단순 지수 함수로 fitting 하는 calTau1 함수와 비선형 함수로 fitting 하는 calTau2 함수를 제공한다. cal_sdip.py 프로그램은 먼저 calTau1 함수로 초기 값을 얻은 후 이 값을 비선형 fitting에 적용하여 결과를 출력한다.

출력된 그림은 파일로 입력한 자료인 Airmass에 따른 파워를 빨간 점으로 표시하고 비선형 결과로부터 유도된 대기 광학적 두께, 이득, 전방 빔 효율 등으로 계산된 모델 값을 선으로 표시해준다.

이 프로그램은 대기의 광학적 두께를 계산해주지만 아래 그림에서 보듯이 이 계산 과정 중에 수신기의 이득과 잡음 온도 안테나의 전방 빔 결합 효율도 부수적으로 출력된다. 이 중 수신기 잡음온도와 전방 빔 결합 효율은 좋은 자료의 경우는 비교적 정확한 값을 얻지만 구름 등의 영향이 있는 경우는 오차가 커지므로 조심해야한다.


```
IPython: kvglib/bin
File Edit View Search Terminal Help
Using matplotlib backend: TkAgg

In [1]:

In [1]: !cat sdip_test.dat
# EL Power
0.0 30.0
18.21 19.3
20.17 18.4
22.62 17.5
25.77 16.6
30.00 15.6
36.03 14.6
45.58 13.6
65.38 12.4

In [2]: execfile('cal_sdip.py')
Data File: sdip_test.dat
Ambient Temp(K): 293
Vane Temp(K): 293
Plot (y/n): y
##### Linear Fitting #####
Tau (EL=90) : 0.23463609370593008
Gain (P/K) : 0.07811560887947563
Trx (K) : 91.0461647849013
Stddev : 0.0052836926632539175
##### Optimize using calTau2 #####
## Initial Conditions
Eta : 0.93
Tatm (K) : 283.0
Tau (EL=90) : 0.23463609370593008
Tspill (K) : 293.0
## Fitting Results
Eta : 1.0106806484988324
Tau (EL=90) : 0.24658166944865684
```

그림 2 ipython에서 대기 투과 프로그램 실행 예

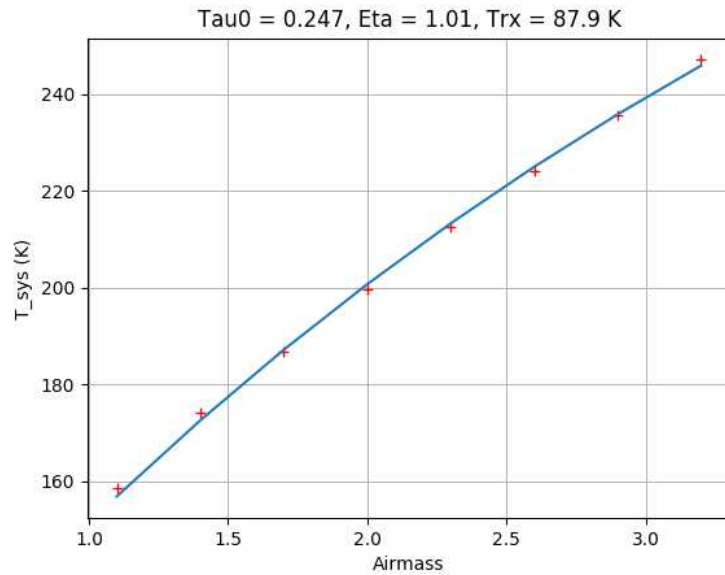


그림 3 대기 투과율 계산 프로그램 해석 결과 Plot

4. 수신기 이득 안정도 측정 프로그램

noise_sp.py와 adev_pl.py는 수신기의 주파수에 따른 이득과 시간에 따른 이득 안정도를 확인하기 위한 프로그램이다. 두 프로그램 모두 시간에 따라 측정된 파워값을 이용해서 계산한다. 어떤 특정 주파수의 파워 변화가 있는 지를 보기위해서는 noise power spectrum을 이용할 수 있다. 아래 그림과 같이 4개의 신호 중 밝은 파란색으로 표시된 신호의 경우 1.2Hz에서 특정한 파워 변화가 있음을 알 수 있다.

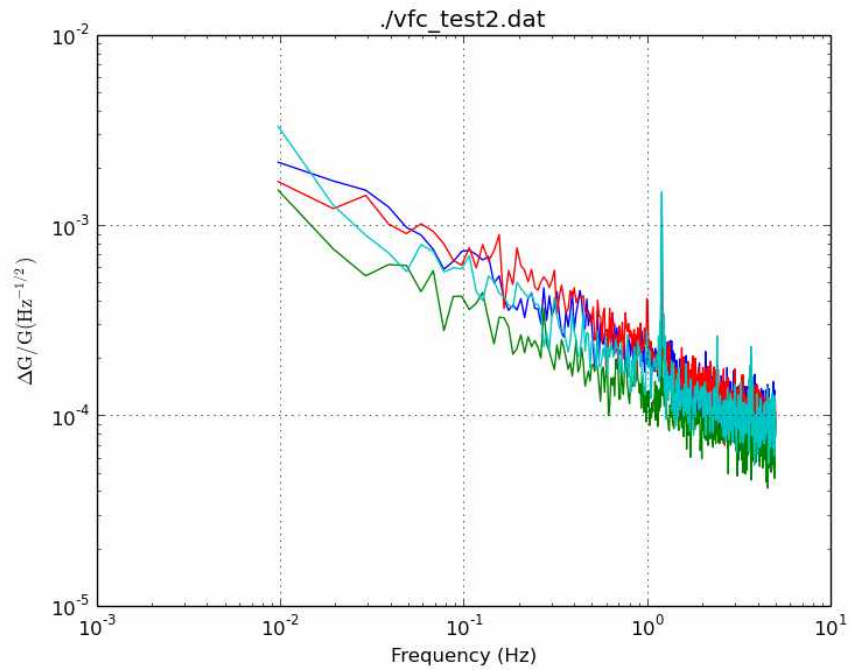


그림 4 수신기 이득 안정도 테스트 : Noise Spectrum Plot

Allan Standard Deviation은 시간에 따라 신호를 평균을 취했을 때 요동값이 얼마나 줄어드는지를 보여준다. 백색잡음이 지배적인 경우는 시간의 0.5 제곱승에 비례해서 줄어든다. 하지만 $1/f$ 잡음이 생기 생기면 거꾸로 0.5 제곱승에 비례해서 요동값이 증가하게 된다. 아래 보여주는 adev_pl.py 로 계산한 Alland Deviation Plot은 요동값이 짧은 시간 영역에서는 기울기가 -0.5인 백색 잡음이 지배적이지만 10초를 넘어서면서는 서서히 이 경향에서 벗어나는 것을 보여준다.

noise_sp.py와 adev_pl.py 모두 샘플링한 자료의 시간 간격과 총 취득 시간에 따라 측정할 수 있는 주파수와 시간대 범위가 결정된다.

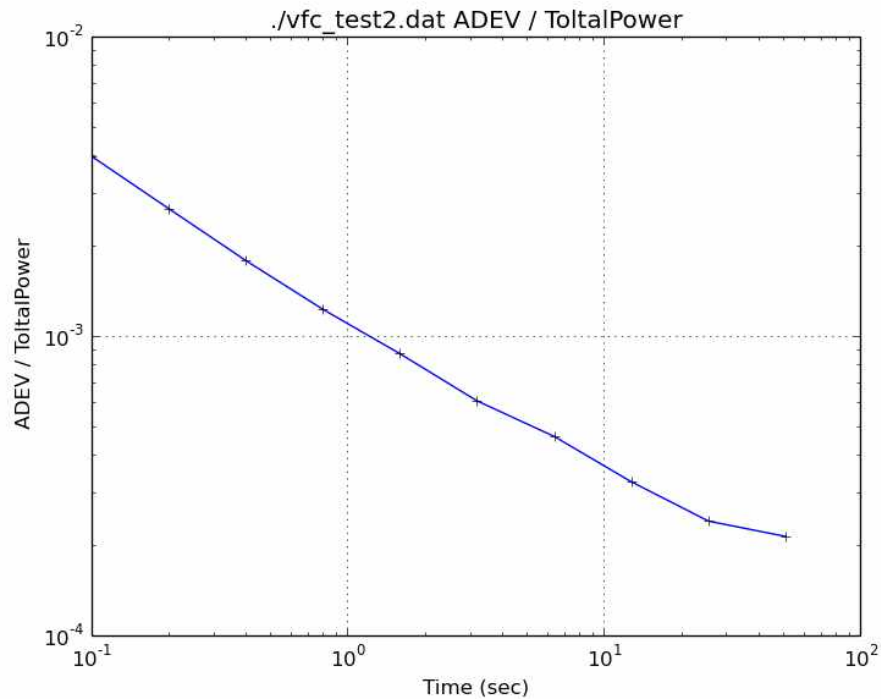


그림 5 수신기 이득 안정도 : Allan Standard Deviation Plot

5. 천체 좌표계산 라이브러리

가. 개요

천체 좌표계산 라이브러리는 Python 언어로 작성되었으며 atropy, astroplan 모듈을 사용한다. 지표면 임의의 위치(위도, 경도, 고도로 정의)에서 관측되는 천체의 정확한 지향과 추적을 위해서 NASA Jet Propulsion Laboratory (JPL)에서 제공하는 Development Ephemeris (DE)와 미해군 관측소의 International Earth Rotation and Reference Systems Service (IERS)에서 제공하는 자료를 사용한다. 이번에 개발된 천체 좌표계산 라이브러리에서는 JPL DE430을 사용하여 1550년부터 2650년까지의 태양계 행성 좌표를 추적할 수 있게 되었다.

천체 좌표계산 라이브러리를 처음 실행하면 사용자 컴퓨터에 JPL DE430과 IERS-A table이 저장되어 있는지 확인하고 없으면 해당 서버에서 자동으로 다운로드하여 좌표계산에 필요한 모듈들을 준비한다. 현재, 설치된 라이브러리는 프로그램 테스트 과정에서 JPL DE430과 IERS-A table이 다운로드 되어 사용자 계정에 저장되어 있다.

나. 설치방법

▪ 사용된 Python 모듈

기본 모듈: numpy, matplotlib, datetime, pytz

천체 좌표계산 모듈: astropy, astroplan, jplephem

▪ Python 모듈 설치방법

시스템에 설치된 기존의 Python 모듈과 충돌을 피하기 위해서 가능하면 super user 권한이 아닌 일반 사용자 계정으로 설치해야 한다. 우분투의 패키지 관리자인 apt-get이나 dpkg 명령어 대신에 Python 패키지 관리자인 pip 명령어를 사용하고 --user 옵션을 사용하여 시스템 디렉토리가 아닌 사용자 디렉토리에 설치되도록 한다.

```
예제
pip install astropy --user
pip install astroplan --user
pip install jplephem --user
```

▪ 천체 좌표계산 라이브러리 설치방법

천체계산 함수가 구현된 coord.py와 세종 관측소와 KVN, VERA 등의 안테나 위치(위도, 경도, 고도)가 정의된 Station.py로 구성되어 있다. coord.py에는 천체 좌표계산에 필요한 함수들이 Coord 클래스의 멤버함수로 정의되어 있다. 두 소스 코드를 사용자 리눅스/맥 환경의 PYTHONPATH가 정의된 디렉토리에 복사하여 사용한다.

다. 천체 좌표계산 라이브러리의 구성과 기능

▪ coord.py의 기능 (Coord 클래스 멤버함수)

set_site(station_code) 사용자 관측위치를 지정한다.	
station_code	Station.py에 정의된 관측소 코드를 사용. 예) Kv=세종, Kt=탐라
dpplr(itime, sra, sdec, slsr, timeflag=None) 천체와 지구의 상대속도에 의한 시선방향 속도성분을 계산한다.	
itime	관측시간. '연-월-일 시:분:초' 형식으로 UTC 기준. timeflag를 사용하지 않을 경우 사용됨. timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다. 예) itime = Time('2019-09-01 02:00:00', scale='utc') * from astropy.time import Time을 미리 선언하여 이용.
sra, sdec	관측 천체의 적경, 적위. 다음과 같은 세 가지 형식이 모두 가능하다. 예)

	<p>sra='03h53m28.927s', sdec='+11d24m21.48s'</p> <p>sra='03:53:28.927', sdec='+11:24:21.48'</p> <p>sra=58.37052917, sdec=11.40596667</p> <p>문자열 입력일 경우 적경은 '시분초', 적위는 '도분초' 단위를 사용하고 숫자값 입력일 경우 적경과 적위 모두 '도' 단위이다.</p>
slsr	관측 천체의 LSR 속도.
timeflag	timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.
g	
EgToGa(sra, sdec) 적도좌표계 (적경, 적위)를 은하좌표계 (은경, 은위)로 변환한다.	
sra, sdec	<p>관측 천체의 적경, 적위. 다음과 같은 세 가지 형식이 모두 가능하다.</p> <p>예)</p> <p>sra='03h53m28.927s', sdec='+11d24m21.48s'</p> <p>sra='03:53:28.927', sdec='+11:24:21.48'</p> <p>sra=58.37052917, sdec=11.40596667</p> <p>문자열 입력일 경우 적경은 '시분초', 적위는 '도분초' 단위를 사용하고 숫자값 입력일 경우 적경과 적위 모두 '도' 단위이다.</p>
GaToEq(sl, sb) 은하좌표계 (은경, 은위)를 적도좌표계 (적경, 적위)로 변환한다.	
sl, sb	<p>관측 천체의 은경, 은위. 다음과 같은 세 가지 형식이 모두 가능하다.</p> <p>예)</p> <p>sl='177d57m17.947s', sb='-31d24m42.647s'</p> <p>sl='177:57:17.947', sb='-31:24:42.647'</p> <p>sl=177.95498516, sb=-31.41184639</p> <p>문자열 입력일 경우 은경과 은위 모두 '도분초' 단위를 사용하고 숫자값 입력일 경우 은경과 은위 모두 '도' 단위이다.</p>
GaToHo(itime, sl, sb, timeflag=None) 은하좌표계 (은경, 은위)를 지평좌표계 (방위각, 고도)로 변환한다.	
itime	<p>관측시간. '연-월-일 시:분:초' 형식으로 UTC 기준. timeflag를 사용하지 않을 경우 사용됨. timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.</p> <p>예) itime = Time('2019-09-01 02:00:00', scale='utc')</p> <p>* from astropy.time import Time을 미리 선언하여 이용.</p>
sl, sb	<p>관측 천체의 은경, 은위. 다음과 같은 세 가지 형식이 모두 가능하다.</p> <p>예)</p> <p>sl='177d57m17.947s', sb='-31d24m42.647s'</p> <p>sl='177:57:17.947', sb='-31:24:42.647'</p> <p>sl=177.95498516, sb=-31.41184639</p> <p>문자열 입력일 경우 은경과 은위 모두 '도분초' 단위를 사용하고 숫자값 입력일 경우 은경과 은위 모두 '도' 단위이다.</p>
timeflag	timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.
g	
star(itime, sra, sdec, timeflag=None) 행성을 제외한 천체 (항성, 퀘이사 등)의 지평좌표계 (방위각, 고도)를 계산한다.	

itime	<p>관측시간. '연-월-일 시:분:초' 형식으로 UTC 기준. timeflag을 사용하지 않을 경우 사용됨. timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.</p> <p>예) itime = Time('2019-09-01 02:00:00', scale='utc')</p> <p>* from astropy.time import Time을 미리 선언하여 이용.</p>
sra, sdec	<p>관측 천체의 적경, 적위. 다음과 같은 세 가지 형식이 모두 가능하다.</p> <p>예)</p> <p>sra='03h53m28.927s', sdec='+11d24m21.48s'</p> <p>sra='03:53:28.927', sdec='+11:24:21.48'</p> <p>sra=58.37052917, sdec=11.40596667</p> <p>문자열 입력일 경우 적경은 '시분초', 적위는 '도분초' 단위를 사용하고 숫자값 입력일 경우 적경과 적위 모두 '도' 단위이다.</p>
timeflag	timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.
<p>planet(itime, sbody, timeflag=None)</p> <p>행성의 지평좌표계 (방위각, 고도) 를 계산한다.</p>	
itime	<p>관측시간. '연-월-일 시:분:초' 형식으로 UTC 기준. timeflag을 사용하지 않을 경우 사용됨. timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.</p> <p>예) itime = Time('2019-09-01 02:00:00', scale='utc')</p> <p>* from astropy.time import Time을 미리 선언하여 이용.</p>
sbody	<p>태양계 행성의 이름.</p> <p>astropy.coordinates.solar_system_ephemeris.bodies에 정의된 이름을 사용한다.</p> <p>예) 'earth', 'sun', 'moon', 'mercury', 'venus', 'mars', 'jupiter', 'saturn', 'uranus', 'neptune'</p>
timeflag	timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.
<p>plotsky(itime, star_name, sra, sdec, sbody, timeflag=None)</p>	
itime	<p>관측시간. '연-월-일 시:분:초' 형식으로 UTC 기준. timeflag을 사용하지 않을 경우 사용됨. timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.</p> <p>예) itime = Time('2019-09-01 02:00:00', scale='utc')</p> <p>* from astropy.time import Time을 미리 선언하여 이용.</p>
star_name	그림에 표시될 천체 (항성, 퀘이사 등)의 이름
sra, sdec	<p>관측 천체의 적경, 적위. 다음과 같은 세 가지 형식이 모두 가능하다.</p> <p>예)</p> <p>sra='03h53m28.927s', sdec='+11d24m21.48s'</p> <p>sra='03:53:28.927', sdec='+11:24:21.48'</p> <p>sra=58.37052917, sdec=11.40596667</p> <p>문자열 입력일 경우 적경은 '시분초', 적위는 '도분초' 단위를 사용하고 숫자값 입력일 경우 적경과 적위 모두 '도' 단위이다.</p>
sbody	태양계 행성의 이름.

	astropy.coordinates.solar_system_ephemeris.bodies에 정의된 이름을 사용한다. 예) 'earth', 'sun', 'moon', 'mercury', 'venus', 'mars', 'jupiter', 'saturn', 'uranus', 'neptune'
timeflag	timeflag='sys'를 사용하면 컴퓨터의 시스템 시간을 사용한다.

■ Station.py에 정의된 사이트 정보

<pre> LocationTable = { 'TN': ['126:27:34.5', '33:17:20.1', 452.6], 'YS': ['126:57:19.1', '37:27:15.6', 200], 'US': ['129:14:59.2', '35:32:43.2', 157], 'SJ': ['127:18:12.09979', '36:31:21.79467', 194.6214], 'MZ': ['141:07:57.31', '39:08:00.68', 116.4], 'IR': ['130:26:23.60', '31:44:52.43', 573.6], 'OG': ['142:12:59.80', '27:05:30.49', 273.1], 'IS': ['124:10:15.59', '24:24:43.82', 65.1], } LocationTable_w_StationCode = { 'Kt': ['126:27:34.5', '33:17:20.1', 452.6], # Tamna 'Ky': ['126:57:19.1', '37:27:15.6', 200], # Yonsei 'Ku': ['129:14:59.2', '35:32:43.2', 157], # Ulsan 'Kv': ['127:18:12.09979', '36:31:21.79467', 194.6214], # Sejong 'Vm': ['141:07:57.31', '39:08:00.68', 116.4], # Mizusawa 'Vr': ['130:26:23.60', '31:44:52.43', 573.6], # Iriki 'Vo': ['142:12:59.80', '27:05:30.49', 273.1], # Ogasawara 'Vs': ['124:10:15.59', '24:24:43.82', 65.1], # Ishigakigima } </pre>	
---	--

라. 천체 좌표계산 라이브러리의 사용 예제

■ 테스트 소스

<pre> import sys, os import numpy as np import astropy.units as u from astropy.time import Time from astropy.coordinates import solar_system_ephemeris </pre>	
---	--


```

import coord
cd = coord.Coord()
cd.set_site('Kv')

#print solar_system_ephemeris.bodies
solar_body = ['earth', 'sun', 'moon', 'mercury', 'venus', \
              'earth-moon-barycenter', 'mars', 'jupiter', \
              'saturn', 'uranus', 'neptune']

star_lsr = [35. * u.km/u.s, 8.5 * u.km/u.s]
star_name = ["IK Tau", "WX Psc"]
star_eq1 = ['03h53m28.927s', '+11d24m21.48s', '01h06m25.988s',
'+12d35m52.88s']
star_eq2 = ['03:53:28.927', '+11:24:21.48', '01:06:25.988',
'+12:35:52.88']
star_eq3 = [58.37052917, 11.40596667, 16.60828333, 12.59802222]
# Select input type
star_eq = star_eq1
#star_eq = star_eq2
#star_eq = star_eq3

print '##### TIME TEST #####'
itime = Time('2019-09-01 02:00:00', scale='utc')
print 'Observation time:', itime
print 'JD:', itime.jd
print 'MJD:', itime.mjd

print '##### IERS_A TEST #####'
cd.iresa_check()

print '##### LST TEST #####'
LST = cd.get_lst(itime)
print 'LST hms:', LST
print 'LST hrs:', LST.hour
print 'LST sec:', LST.hour*3600.

print '##### DOPPLER TEST #####'
print star_name[0]
vlsr = cd.dpplr(itime, star_eq[0], star_eq[1], star_lsr[0])
print vlsr
print vlsr.value
print '-----'

```

```

print star_name[1]
vlsr = cd.dpplr(itime, star_eq[2], star_eq[3], star_lsr[1])
print vlsr
print vlsr.value
print '##### CD TEST #####'
print 'EQ to GA'
ga = cd.EqToGa(star_eq[0], star_eq[1])
print 'in EQ (hms dms):', star_eq[0], star_eq[1]
print 'out GA (dms dms):', ga[0], ga[1]
print 'out GA (deg deg):', ga[0].degree, ga[1].degree
print '-----'
print 'GA to EQ'
eq = cd.GaToEq(ga[0], ga[1])
print 'in GA (dms dms):', ga[0], ga[1]
print 'out EQ (dms dms):', eq[0], eq[1]
print 'out EQ (deg deg):', eq[0].degree, eq[1].degree
print '-----'
print 'GA to HO'
ho = cd.GaToHo(itime, ga[0], ga[1])
print 'in GA (dms dms):', ga[0], ga[1]
print 'out HO (dms dms):', ho[0], ho[1]
print 'out HO (deg deg):', ho[0].degree, ho[1].degree

print '##### STAR TEST #####'
print star_name[0]
outcoord = cd.star(itime, star_eq[0], star_eq[1])
#print 'Az El:', outcoord
print 'Az El (dms dms):', outcoord[0], outcoord[1]
#print 'Az El (dms dms):', outcoord[0].dms, outcoord[1].dms
#print 'Az El (hms dms):', outcoord[0].hms, outcoord[1].dms
print 'Az El (deg deg):', outcoord[0].degree, outcoord[1].degree
print '-----'
print star_name[1]
outcoord = cd.star(itime, star_eq[2], star_eq[3])
print 'Az El (dms dms):', outcoord[0], outcoord[1]
print 'Az El (deg deg):', outcoord[0].degree, outcoord[1].degree

print '##### PLANET TEST #####'
sbody = solar_body[7]
print sbody
outcoord = cd.planet(itime, sbody)
#print 'RA Dec:', outcoord

```

```

print 'RA Dec (dms dms):', outcoord[0], outcoord[1]
#print 'RA Dec (dms dms):', outcoord[0].dms, outcoord[1].dms
#print 'RA Dec (hms dms):', outcoord[0].hms, outcoord[1].dms
print 'RA Dec (deg deg):', outcoord[0].degree, outcoord[1].degree
#print 'Az El:', outcoord
print 'Az El (dms dms):', outcoord[2], outcoord[3]
#print 'Az El (dms dms):', outcoord[2].dms, outcoord[3].dms
#print 'Az El (hms dms):', outcoord[2].hms, outcoord[3].dms
print 'Az El (deg deg):', outcoord[2].degree, outcoord[3].degree

print '##### ALT PLOT #####'
cd.plotsky(itime, star_name[0], star_eq[0], star_eq[1], solar_body[7])
#cd.plotsky(itime, star_name[0], star_eq[0], star_eq[1], '')
#cd.plotsky(itime, '', star_eq[0], star_eq[1], solar_body[7])

```

■ 테스트 소스 출력결과

```

Selected site: Kv
<Observer: name='Kv',
      location (lon, lat, el)=(127.303361053 deg, 36.5227207417 deg,
194.6214 m),
      timezone=<UTC>,
      pressure=<Quantity 0. bar>,
      temperature=<Quantity 0. deg_C>,
      relative_humidity=0.0>
##### TIME TEST #####
Observation time: 2019-09-01 02:00:00.000
JD: 2458727.58333
MJD: 58727.0833333
##### IERS_A TEST #####
IERS_A cached
Cache dir: /home/yjyun/.astropy/cache/download/py2
##### LST TEST #####
LST hms: 9h09m00.87s
LST hrs: 9.15024165605
LST sec: 32940.8699618
##### DOPPLER TEST #####
IK Tau
18.9727887921 km / s
18.9727887921
-----
WX Psc

```

```

-8.17095773369 km / s
-8.17095773369
##### CD TEST #####
EQ to GA
in EQ (hms dms): 03h53m28.927s +11d24m21.48s
out GA (dms dms): 177d57m17.9466s -31d24m42.647s
out GA (deg deg): 177.954985156 -31.4118463912
-----
GA to EQ
in GA (dms dms): 177d57m17.9466s -31d24m42.647s
out EQ (dms dms): 58d22m13.905s 11d24m21.48s
out EQ (deg deg): 58.3705291667 11.4059666667
-----
GA to HO
in GA (dms dms): 177d57m17.9466s -31d24m42.647s
out HO (dms dms): 272d39m24.3616s 15d53m10.1192s
out HO (deg deg): 272.656767109 15.8861442133
##### STAR TEST #####
IK Tau
Az El (dms dms): 272d39m24.3616s 15d53m10.1192s
Az El (deg deg): 272.656767109 15.8861442133
-----
WX Psc
Az El (dms dms): 299d12m13.7555s -15d24m28.5418s
Az El (deg deg): 299.203820976 -15.4079282834
##### PLANET TEST #####
jupiter
RA Dec (dms dms): 253d37m39.2096s -22d14m22.3542s
RA Dec (deg deg): 253.62755822 -22.2395428202
Az El (dms dms): 93d58m09.6697s -34d00m18.8115s
Az El (deg deg): 93.9693526821 -34.0052254029
##### ALT plot #####

```

부록 1. KVGCS 소프트웨어 설치 및 설정 파일

1. 소프트웨어 구성 및 디렉토리 구조

다음 우주측지센터 관측 SW의 주요 디렉토리 구조이다.

표 1 KVGCS SW 디렉토리 구조 및 용도

디렉토리 명				디렉토리 용도 구분
/home/kvg	/KVGCS	/Main	/kvgcs(*)	주 관측 프로그램
		/Antenna		안테나 제어 프로그램
		/AWS		기상 자료 획득 프로그램
		/wxGUI		GUI 프로그램
		/DataServer		자료획득 서버 프로그램
		/Data	/DSM	분광자료 디렉토리
			/VFC	Total Power 자료 디렉토리
			/EXPER	VLBI 관측 Calibration 단일 경 자료 디렉토리
		/Resource	/Vex	Vex 파일
			/Catalog	천체 좌표 목록
			/Skd	Skd 파일
			/Config	시스템 설정 파일
			/Ephem	천체 좌표 계산 파일
			/Station	망원경 좌표
			/GainCurve	망원경 이득 곡선 측정값
		/Log	/VLBI	Station Log
			/KCS	kvgcs 제어 통신 로그
			/TY	ANTAB 파일
			/Observation	관측 로그
			/AWS	기상 로그
			/PointingModel	Pointing Model 계산 결과
			/Hexapod	부경 모델 측정
			/DBBC3	DBBC3 제어 로그
		/bin		장치 제어 프로세스 또는 보조 프로그램 실행
		kvgrc		환경변수 설정 파일

KVGCS각 가장 상위 디렉토리에 해당하며 KVGCS 디렉토리 아래에 기능에 따라 구분된 하위 디렉토리가 있다. Main 디렉토리 아래에 있는 kvgcs 디렉토리가 주 관측 프로그램의 디렉토리이다.

Antenna, AWS, DataServer는 각각의 장치 제어 프로그램의 디렉토리이다. Data, Resource, Log는 각각 관측 자료와 시스템 설정과 보조 자료들, 로그 파일을 저장하기 위한 디렉토리이다. bin 디렉토리 아래에는 KVGCS의 관련 프로세스를 시동/종료 시키는 스크립트와 여러 보조 프로그램들이 있다.

2. 환경 설정

kvgrc 파일은 관측 프로그램에서 필요로하는 여러 환경 변수가 설정되어있다. ~/.bashrc에서 kvgrc 를 자동 실행하도록 설정해두었다.

```
export KVGHOME=$HOME/KVGCS
export KVG_RESOURCE=$HOME/KVGCS/Resource
export KVG_LOG=$HOME/KVGCS/Log
export KVG_DATA=$HOME/KVGCS/Data
export PYTHONPATH=$KVGHOME:$KVGHOME/Main/kvgcs/:$KVGHOME/Loc:
    /usr/lib/python2.7/dist-packages/wx-2.8-gtk2-unicode

export KCS_HOST='MPC'
export KCS_STATION='KVGSI'
export KCS_STATION_CODE='KSI'
export KCS_HOME=$KVGHOME
export KCS_WX=$KVGHOME/wxGUI
export KCS_CONFIG=$KVG_RESOURCE/Config
export KCS_CATALOG=$KVG_RESOURCE/Catalog
export KCS_RESOURCE=$KVG_RESOURCE
export KCS_VEX=$KVG_RESOURCE/Vex
export KCS_XYZ=$KVG_RESOURCE/Station
export KCS_LOG=$KVG_LOG
export KCS_KLOG=$KVG_LOG/KCS
export KCS_VLBLOG=$KVG_LOG/VLBI
export KCS_SCRIPT=$KVGHOME/Main/kvgcs/etc/scripts/
export KCS_HAS_DIO='False'
export KCS_DATA=$KVG_DATA
export KCS_PTD=$KCS_DATA/ptd

export PATH=$PATH:$KVGHOME/bin
```

우주측지 시스템의 장치 제어를 위해 구축한 사설망의 정보는 Resource/Config/device.conf 파일에 정의되어있다. 이 파일을 편집해서 필요한 장치 제어 프로세스와 포트 번호 주 관측프로그램과의 연결 여부를 제어할 수 있다. device.conf 파일의 일부 설정을 보면 아래와 같다. 일부를 제외한 대부분 장치들은 사설망을 통해 연결된다. IP 주소는 보안을 위해 모두 xxx.xxx.xxx.xxx 로 표기하였다.

[Antenna]

IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	60110
MonPort	=	60111
OpMode	=	'On'
MonPeriod	=	0.5

[Clock]

IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	60030
MonPort	=	60030
OpMode	=	'On'
MonPeriod	=	0.5

[Rx]

IP	=	' 'xxx.xxx.xxx.xxx'
CtrlPort	=	5005
MonPort	=	5005
OpMode	=	'On'
MonPeriod	=	0.5

[IFSelector]

IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	9980
MonPort	=	9980
OpMode	=	'On'
MonPeriod	=	0.5

[VideoConverter1]

IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	9981
MonPort	=	9981
OpMode	=	'On'
MonPeriod	=	0.5

[VideoConverter2]

IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	9982
MonPort	=	9982
OpMode	=	'On'
MonPeriod	=	0.5

[DataServer]

IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	60060
MonPort	=	60061
OpMode	=	'On'
MonPeriod	=	0.5

[Aws]

IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	60050

MonPort	=	60051
OpMode	=	'On'
MonPeriod	=	0.5

[MK6]		
IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	14242
MonPort	=	14242
OpMode	=	'On'
MonPeriod	=	0.5

[ObsMode]		
IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	60162
MonPort	=	60163
OpMode	=	'Off'
MonPeriod	=	0.5

[Scheduler]		
IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	60160
MonPort	=	60161
OpMode	=	'On'
MonPeriod	=	0.5

[DBBC3]		
IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	4000
MonPort	=	4000
OpMode	=	'On'
MonPeriod	=	0

[GSM]		
IP	=	'xxx.xxx.xxx.xxx'
CtrlPort	=	60100
MonPort	=	60100
OpMode	=	'On'
MonPeriod	=	0.5

각 장치제어 프로세스는 제어와 감시 2개의 포트 연결을 지원 하는 것을 가정한다. 경우에 따라 하나의 접속 포트를 가지는 경우 제어 포트와 감시 포트를 동일 번호를 설정할 수 있다. 이 경우 OpMode 를 Off 로 설정한 경우 kvgcs 에서 장치 연결을 하지 않는다. 테스트 시 특정 장비의 연결을 제외할 경우 OpMode를 Off로 변경한 뒤에 kvgcs를 실행시킨다. OpMode를 Off로 선택한 경우는 kvgcs에서 IP 주소에 대한 정보도 등록되지 않는다. 이 후 수동으로 연결을 하기 위해서는 IP 주소 정보를 수동으로 설정해줘야 한

다.

kvcgs에서 현재 등록된 장치와 주소, 제어와 상태 감시 연결 상태를 확인하는 다음 함수를 지원한다.

```
tci.tcp_status()
tmi.tcp_status()
```

개별 장비의 통신 연결을 끊고 연결하기 위해서는 다음 명령을 사용한다.

```
tci.connect(DeviceName)
tmi.connect('DeviceName)
tci.close(DeviceName)
tmi.close(DeviceName)
```

여기서 DeviceName은 device.conf 파일에 등록된 이름이다.

부록 2. DBBC3 운영 매뉴얼



그림 1 세종 우주측지 센터 DBBC3

우주측지센터에서 VLBI 관측에 사용하던 K5 터미널은 16개 채널의 아날로그 Video DownConverter와 16개의 샘플러 모듈, 샘플러로부터 USB 포트로 자료를 입력 받은 하드디스크 기반의 기록 PC로 구성되어 있다. 16개의 독립적인 Video Converter는 독립적인 위상 오프셋을 주기 때문에 각 채널간의 위상 차이를 보정하기 위한 P-CAL 신호가 필요하다. 낮은 샘플링 주파수와 USB 통신 방식에 따른 기록 속도의 제한 등으로 K5 터미널의 성능은 1Gbps의 관측까지만 가능하다.

최근 디지털 기술이 발달하면서 초고속 샘플러가 널리 도입되고 있고 VLBI 장비 간 통신도 광케이블을 이용하여 기존 K5 터미널보다 16-32배 높은 사양의 디지털 백엔드가 많이 사용되고 있다.

세종 우주측지센터에서 보유하고 있는 DBBC3는 2개의 Conditioning Module (GCoMo)와 2개의 ADB3L 샘플러 보드, 2개의 Core3H 자료 처리 보드로 구성되어 있다. ADB3L 보드는 4개의 샘플러로 구성되어 있고 이 샘플러는 2048MHz로 샘플링이 가능하다. 세종의 ADB3L은 4개 샘플러에 90도 위상차를 줘서 최대 8192MHz 샘플링 자료까지 생성할 수 있다. ADB3L의 4개의 샘플러를 독립적인 4개 입력신호를 처리하거나 2개씩 그룹을 묶어서 2개의 입력 신호를 처리할 수도 있지만 우주측지센터가 도입한 ADB3L는 하나의 입력 신호에 대해 광대역 처리만 지원하는 보드이다. ADB3L 출력 신호는 128Mbit/s로

Core3H 보드로 전달된다.

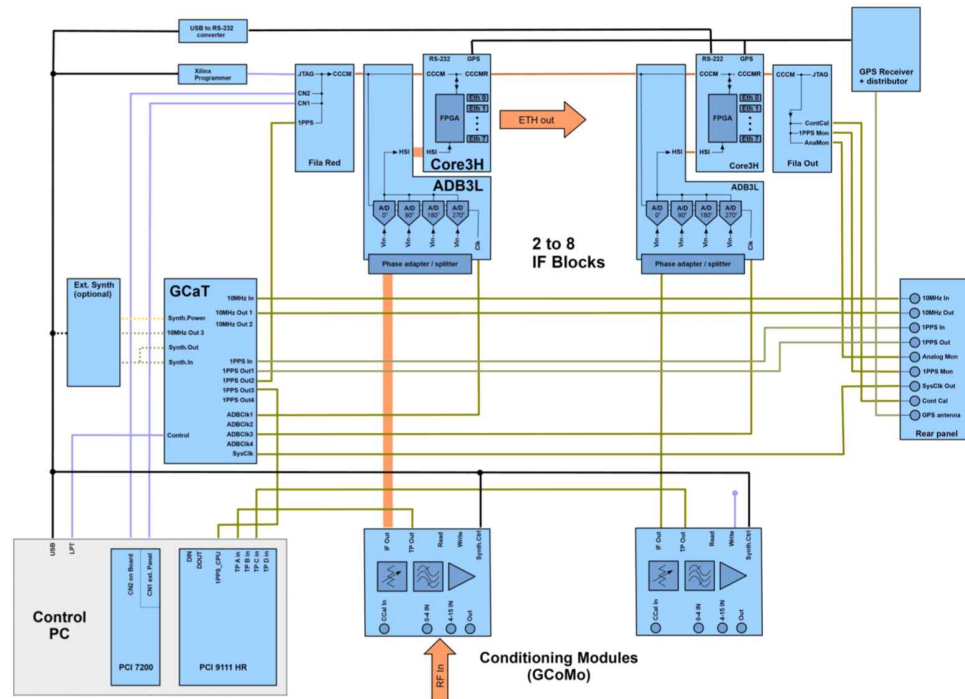


그림 2 DBBC3 구성도

Core3H 보드는 Digital Down Conversion 기능과 신호 양식을 변환하여 10GbE 광케이블로 출력하는 기능을 지원한다. Digital Down Conversion은 최대 8개의 USB, LSB 대역쌍에 해당하는 16개 채널의 출력을 지원한다. Core3H는 4개의 10GbE 출력이 가능한데 현재 Core3H 보드 하나 당 2개의 광케이블만 연결되어 있다.

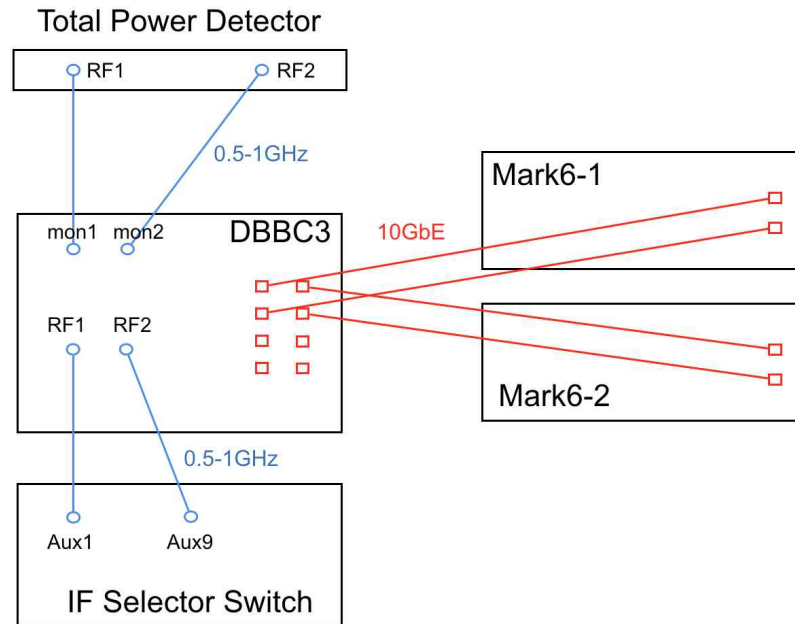


그림 3 우주측지센터 DBBC3와 주변 장치 연결

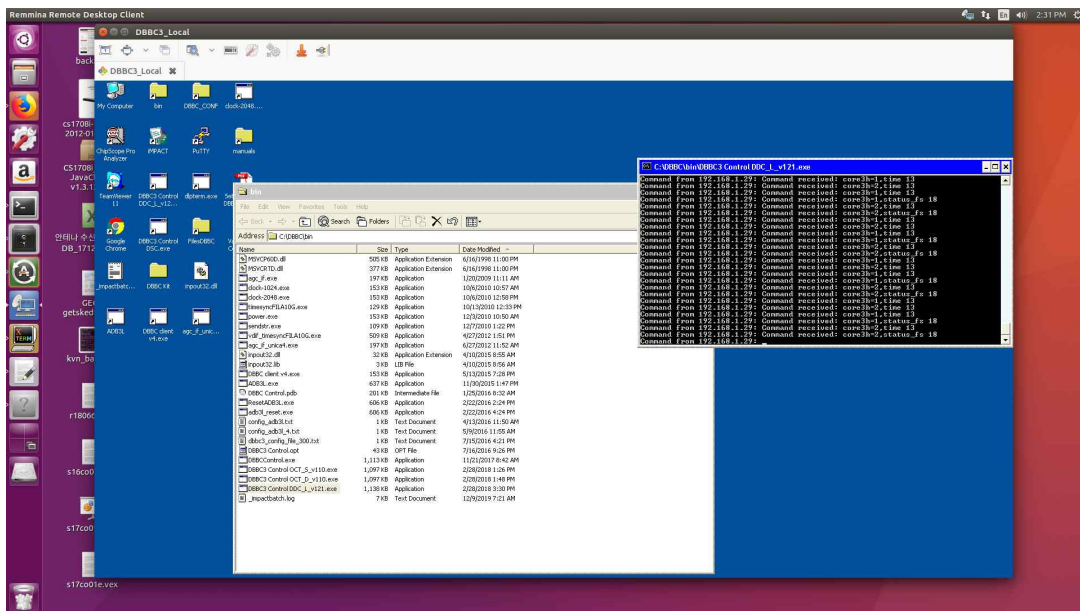


그림 4 DBBC3 운영 Windows 7 창

DBBC3는 Windows 컴퓨터를 내장하고 있어 이 컴퓨터에서 돌아가는 소프트웨어를 통

해 직접 DBBC3를 제어할 수 있다. 이 소프트웨어는 시작 시 설정 파일들을 읽어서 기본적인 설정을 할 수 있게 되어있다. 또한 원격 제어를 위해 TCP/IP 통신을 통한 원격 명령어 제어 입력을 지원한다.

DBBC3 제어 컴퓨터는 2개의 네트워크 포트를 지원한다.

이 중 하나는 내부 제어 사설망 IP를 할당하였고 다른 하나는 외부에서 유지 보수 혹은 서비스를 위해 접속할 수 있도록 외부망 IP를 할당해놓았다. 외부망의 케이블 연결은 보상을 위해 평상시는 차단되어 있으며 필요가 있을 경우 한시적으로 연결한다. 관측용 컴퓨터에서는 원격데스크탑 프로그램인 remmina를 이용해서 DBBC3 제어 컴퓨터의 화면을 관측용 컴퓨터에 불러올 수 있다.

세종 DBBC3의 사양을 정리하면 다음 표와 같다.

표 1 우주측지센터 DBBC3 구성 현황 및 주요 사양

	주요 사양
GCoMo	<ul style="list-style-type: none"> • 32dB Programable Attenuator • Total Powe over full band • Manual or Automatic Gain Control • Direct 0-4GHz input or Downconverter 4-15GHz
ADB3L	<ul style="list-style-type: none"> • 1 x 0-4GHz input • 2048MHz Sampling Clock • 16x8-bit Output Data
Core3H	<ul style="list-style-type: none"> • Digital Down Conversion up to 16 BBC • Fila10G (timesync, packetizing) • 16bus x 8 bit • Output Rates : 2, 4, 8 or 16Gbps • 4 Optical Interfaces

DBBC3는 Firmware를 선택하기에 따라 DSC, DDC_L, DDC_V, OCT_S, OCT_D 모드로 사용할 수 있다. 각 모드 별 지원하는 사양은 아래 표와 같다. 이 중 DDC_L 모드가 현재 측지 관측에서 주로 사용하는 8MHz, 16MHz 대역 지원한다.

KVGCS 소프트웨어에 포함된 DBBC3 제어/감시 소프트웨어는 DDC_L의 기능 사용에 초점이 맞춰져있다. 다른 OCT 모드에 대해서는 추가 개발이 필요한 상황이다. DBBC3를 관측에 활용하기 위해서는 DBBC3를 구성하는 각 보드의 기능을 제어하고 상태를 확인하는 기능이 구현되어야한다. DBBC3 소프트웨어는 최상위 DBBC 제어 소프트웨어를 통해 Core3H, ADB3L, Synthesizer 등 내장 보드를 제어하도록 되어있다. 각 보드를 제어하는 명령어와 명령어 양식이 따로 정의가 되어있으며 명령어 체계가 통일되어있지 않

기 때문에 각각 개별 인터페이스 모듈을 개발해야한다. 이를 위해 KVGCS에서의 인터페이스 모듈도 DBBC3 원격 제어를 위해 DBBC3.py, ADB3L.py, Core3H.py로 모듈화 하였다.

표 2 DBBC3 firmware 별 지원 기능

	Firmware 별 지원 성능
DSC	<ul style="list-style-type: none"> • Direct Sampling Conversion • Full 4GHz bandwidth/IF
DDC_L (Legacy)	<ul style="list-style-type: none"> • Digital Tunable DownConversion • 16 BBCs/IF with fully tunable frequency • BW : 2, 3, 8, 16, 32MHz
DDC_V (VGOS)	<ul style="list-style-type: none"> • 32MHz filters • 8 BBCs/IF
DDC_U (Unified)	<ul style="list-style-type: none"> • BW : 1,2,4,8,16,32,128 MHz • Up to 16 BBCs/IF • under development
OCT_S	<ul style="list-style-type: none"> • Single 32-tap FIR-Filter/IF • 512MHz BW : 0-512, 512-1024, .. 3584-4096 • 1024MHz BW : 0-1024, .. ,3072-4096 • 2048MHz BW : 0-2048, 2048-4096
OCT_D	<ul style="list-style-type: none"> • Double 32-tap FIR-Filter/IF • 512MHz BW : 0-512, 512-1024, .. 3584-4096 • 1024MHz BW : 0-1024, .. ,3072-4096 • 2048MHz BW : 0-2048, 2048-4096

부록 3. 이메일 이벤트 알림 설정 매뉴얼

관측진행 도중 다양한 이벤트가 발생할 때 이를 구분하여 관련자에게 메일을 자동으로 전송하고 현황을 웹메일 환경에서 확인할 수 있도록 시스템을 구현하여 장애 발생 시 신속 대응 체계를 마련한다. 메일 송신 프로그램은 구글의 gmail을 이용하여 1일 최대 500건을 발송할 수 있다.

1. 이벤트 알림 프로그램 흐름도

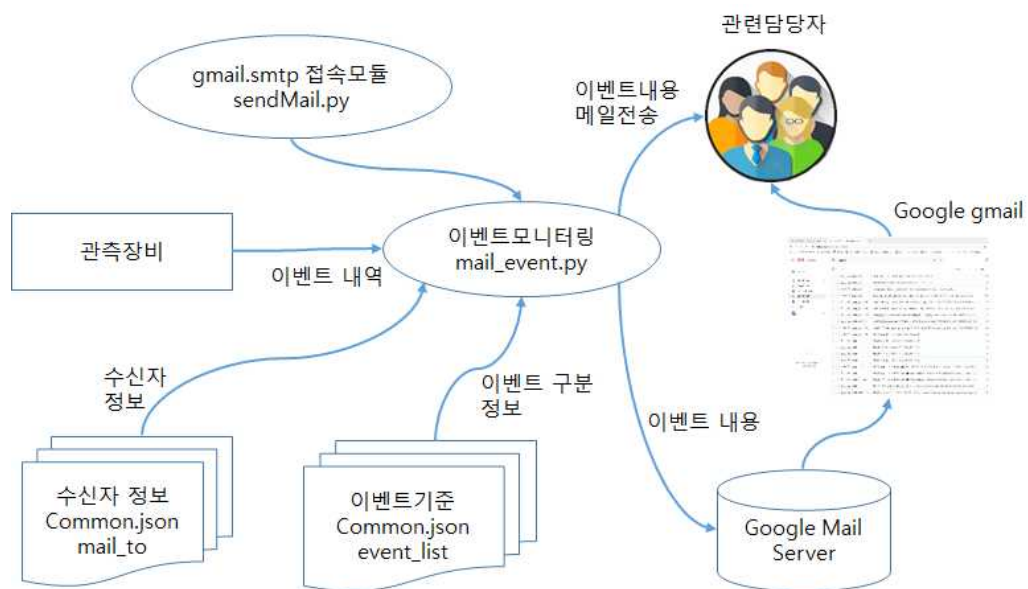


그림 1 이벤트 알림 프로그램 흐름도

가. 서버 및 개발 환경

표 1 이벤트 알림 서비스 서버 및 개발 환경

기능	설명
OS	Ubuntu 16.04.3 LTS
웹서버	Apache/2.4.18 (Ubuntu)
개발환경	Python: Python 2.7.12

나. 이벤트 알림 프로그램 설명

gmail을 사용하기 위해서는 Google 계정이 필요하며 브라우저에서 메일내용을 확인할 수 있다. 특정한 서버 기기에서 메일을 전송하기 위해서는 IMAP설정과 별도의 앱 비밀번호를 추가 생성하여 프로그램에 설정해야 한다.

- 1) gmail 계정에 로그인(gmail 계정 생성 시 설정한 비밀번호 사용)
- 2) 우측 상단에 아래 사진과 같이 톱니바퀴 모양의 아이콘 클릭

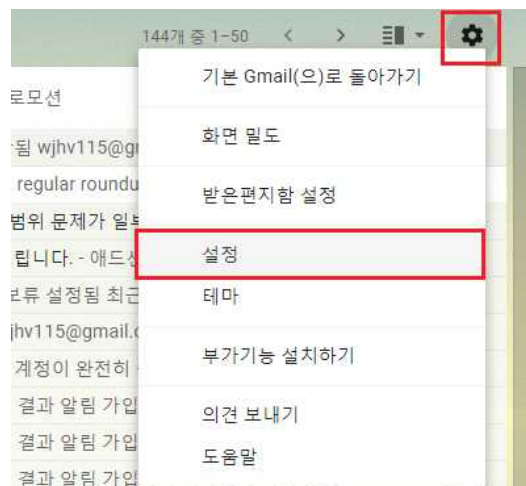


그림 2 이벤트 알림 gmail 설정

- 3) 상단의 카테고리중 '전달 및 POP/IMAP' 카테고리를 선택하고 IMAP 액세스에서 IMAP 사용 클릭.

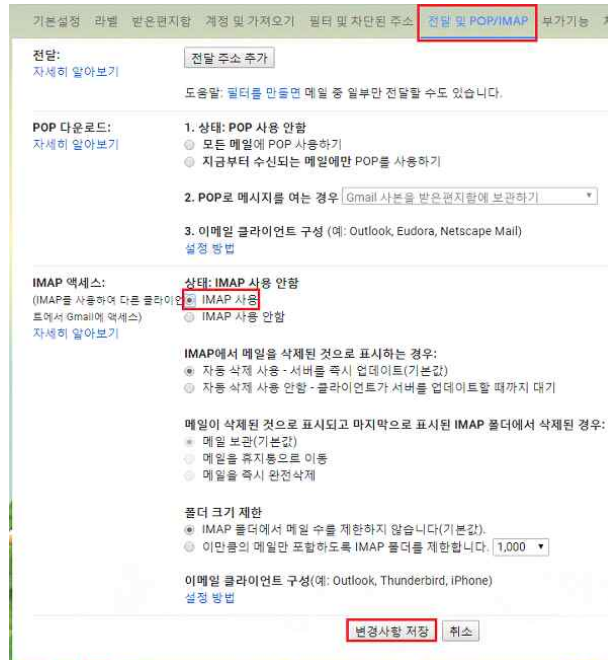


그림 3 이벤트 알림 IMAP 설정

4) 변경사항 저장 (IMAP을 설정하지 않을 경우 로그인 시도 차단됨)

gmail 화면설정에서 위 과정을 거친다음 Google계정 설정에서 보안을 선택하여 앱 비밀번호를 얻어야 한다. 2단계 인증을 완료해야하며 (휴대폰 인증 활용)

2단계 인증이 끝나고 나면 앱 비밀번호를 생성할 수 있는 화면이 표시된다.

5) 앱 비밀번호 생성하기

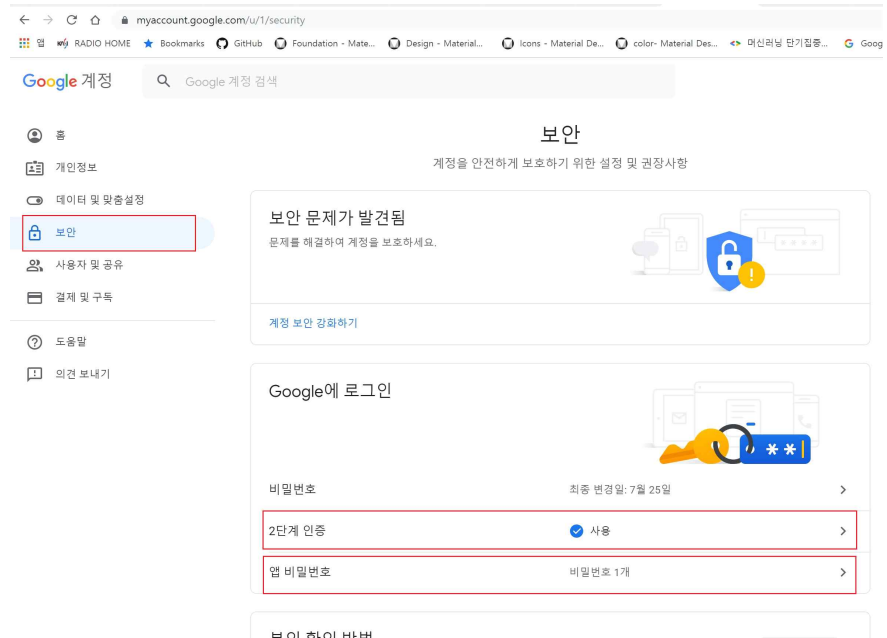


그림 4 이벤트 알림 보안 설정



그림 5 이벤트 알림 비밀 번호 설정

생성된 앱 비밀번호

Windows 컴퓨터용 앱 비밀번호



사용 방법

1. '메일' 앱을 엽니다.
2. '설정' 메뉴를 엽니다.
3. '계정'을 선택한 뒤 '내 Google 계정을 선택'합니다.
4. 비밀번호를 위해 표시된 16자리 비밀번호로 교체합니다.

일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되므로 적어 놓거나 다른 사용자와 공유하지 마세요.

[자세히 알아보기](#)

[확인](#)

그림 6 이벤트 알림 google account 설정

앱 선택은 메일을 선택하고, 기기선택은 기타(맞춤이름)을 선택 후 linux(임의의 이름)를 입력하여 생성버튼을 클릭하면 노란색 박스에 앱 비밀번호(16자리)가 생성된다. 화면이 전환되면 비밀번호를 다시 확인할 수 없으므로 생성된 비밀번호를 복사해 둔다. 로그인 인증부분의 앱 비밀번호에 적용해야 한다.

2. 이벤트 전송 함수

sendMail.py 모듈은 메일을 전송하기 위한 함수이다. 이벤트가 발생하는지 모니터링하는 모듈에서 메시지를 전송할 때 import하여 사용한다.

```
## 프로그램 시작
#-*- coding:utf-8 -*-
import os
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEBase import MIMEBase
from email.MIMEText import MIMEText
from email.header import Header
from email import Encoders
from datetime import datetime

#gmail 계정
```

```

gmail_user = 'vlbisejong@gmail.com'

#gmail 계정(웹에서 접근할때) password -> *****
#메일전송을 프로그램으로 할때 smtp에 2단계 인증의 앱 비밀번호 사용
gmail_pwd = '*****'

#gmail 닉네임
gmail_username='vlbisejong'

# mail_to : 수신자이메일 list
# subject : 제목
# var_msg : 본문

def send_gmail(mail_to, subject, var_msg):
    msg = MIMEText(var_msg, 'plain', 'utf-8')
    # 본문 plain text 형식인 경우
    msg['From']=gmail_username
    msg['To']=",".join(mail_to)
    # 수신자가 list인 경우 ,로 구분

    msg['Subject']=Header(subject,'utf-8')          # encoding 제목

    mail_smtp = smtplib.SMTP('smtp.gmail.com', 587)
    # 세션 생성 SMTP 인스턴스를 이용하여 SMTP 연결을 캡슐화
    # 첫번째 매개변수는 지메일을 사용하기 위한 SMTP 변수이며,
    #두번째 매개변수는 포트번호 지메일의 경우 포트번호 587을 사용

    mail_smtp.ehlo()                                # 서버 연결 테스트
    mail_smtp.starttls()                             # TLS 보안 시작
    mail_smtp.ehlo()                                # 서버 연결 테스트
    mail_smtp.login(gmail_user, gmail_pwd)           # 로그인 인증
    mail_smtp.sendmail(gmail_user, mail_to, msg.as_string()) # 메일발송
    mail_smtp.quit()    # 세션종료
    ## 프로그램 종료

```

3. 이벤트 전송

이벤트가 발생하면 sendMail.py를 import하여 필요한 모듈을 생성하여 사용한다. 아래는 메일 테스트를 위한 프로그램 예시 이다.(mail_test.py)

```

## 프로그램 시작
#-*- coding:utf-8 -*-
from sendMail import *
import json
from datetime import datetime

with open('../comm/common.json') as data_file:

```

```

json_data = json.load(data_file)

# 수신자 리스트
list_mail_to = json_data["mail_to"]["all"]

# 이벤트 종류
list_event_type = json_data["event_list"]["all"]

# 이벤트 종류 중 하나의 값만 선택하여 전송
var_event_type = list_event_type[3].encode('utf-8')

#메일 제목
var_subject = "vlbi test " + var_event_type + " 발생 "

now = datetime.now()
#이벤트 발생시간
event_time = now.strftime('%Y-%m-%d %H:%M:%S')

#메일 내용
var_msg = '''
vlbi운영중 %s 에러가 %s 시에 발생했습니다. 확인바랍니다.
이메시지는 시스템에서 자동발송하였습니다.
''' % (var_event_type, event_time)

send_gmail(list_mail_to, var_subject, var_msg)
## 프로그램 종료

```

4. 수신자와 이벤트 리스트

/home/oper/www/comm/common.json는 수신자 정보와 이벤트 기준 정보를 저장하는 json 포맷으로 mail_to 키는 메일수신자를, event_list 키는 장비운영 시 발생하는 이벤트 코드를 배열로 저장하고 있다.

mail_to 객체내에 다시 obs, equi, all로 정의되어 있으며 obs는 관측관련자 명단을 배열로 가지고 있다. equi는 equipment약자로 장비관련자 명단을 배열로 가지도록 정의하였다. all은 obs와 equi 명단을 통합하여 모아놓은 배열로 구성하여 메일수신자 전체를 지정할 수 있도록 구조를 정의하였다.

event_list 객체는 obs, equi, all로 구분하였으며 obs는 관측관련 이벤트를 배열로 구성하고, equi는 장비관련 이벤트를, 그리고 all은 obs와 equi 이벤트를 통합하여 모아놓은 이벤트 배열이다.

```

{
  "mail_to": {
    "obs": [
      "jsshin@kasi.re.kr",
      "byundy90@gmail.com",
    ],
    "equi": [
      "jsshin@kasi.re.kr",
      "byundy90@gmail.com",
    ],
    "all": [
      "jsshin@kasi.re.kr",
      "byundy90@gmail.com",
    ]
  },
  "event_list": {
    "obs": [
      "Late start of observation",
      "Severe weather condition",
      "Tsys high",
      "Pointing problems",
      "Scheduling related problems",
    ],
    "equi": [
      "Problems with recorder",
      "Antenna control failures",
      "Receiver fault",
      "Receiver fault",
      "DAS/Clock errors",
      "error",
    ],
    "all": [
      "Late start of observation",
      "Problems with recorder",
      "Severe weather condition",
      "Tsys high",
      "Antenna control failures",
      "Receiver fault",
      "Pointing problems",
      "Receiver fault",
      "Scheduling related problems",
      "DAS/Clock errors",
      "error",
    ]
  }
}

```

5. 이벤트 전송 현황 보기

Google계정: vlbisejong@gmail.com 으로 접속하여 gmail 화면의 보낸편지함에서 확인할 수 있다.

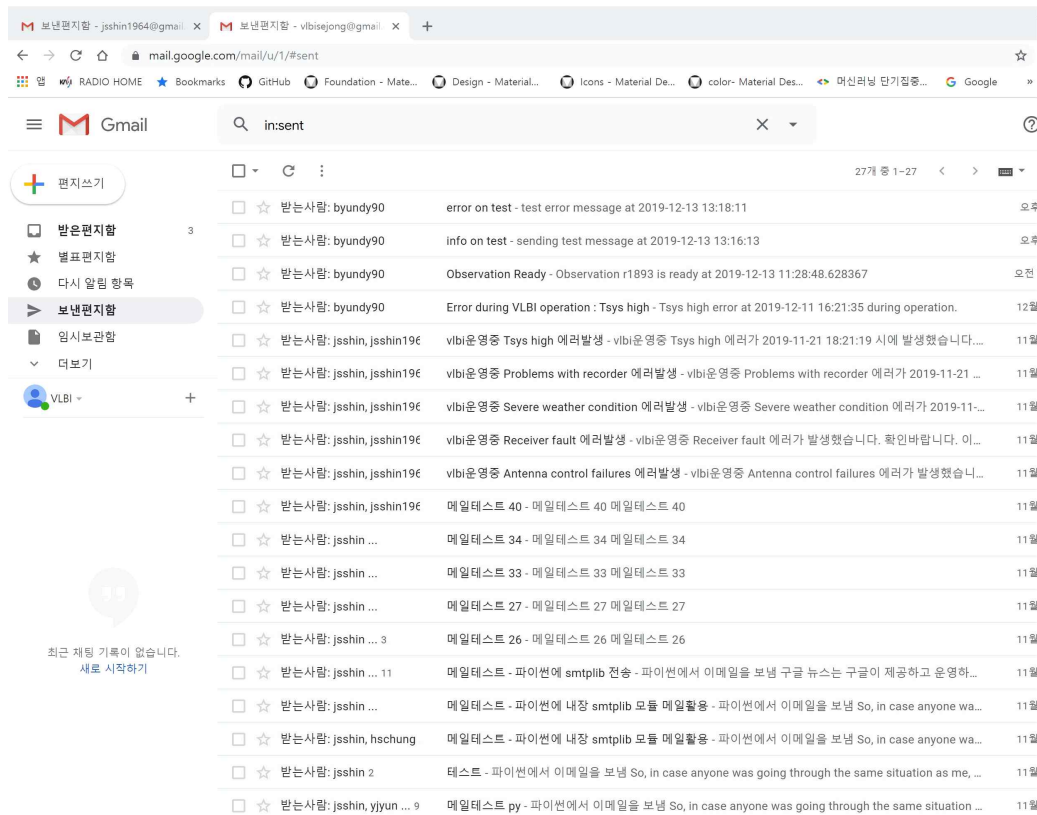


그림 7 gmail을 이용한 이벤트 전송 결과 예시

부록 4. Station Log 양식

관측 진행 상황과 관측 중 기상 상태, 시스템 성능을 대표하는 등가 잡음 온도 등은 관측 중 관측 로그에 자동으로 기록한다. 앞서 언급하였듯이 VLBI 관측 중 관측 진행 상황과 관련된 Station가 기록되며 이 Station Log는 나중에 상관처리와 기선해석을 위해 사용된다. 이 Station Log는 Clock 정보, 관측 천체, 기상 상태, 등가 잡음 온도, 주파수 설정, 기록 디스크팩, 에러 여부가 기록 된다. Station Log는 VLBI 관측 스크립트인 runKVex_DBBC3.py가에서 만들어지며 Station Log 생성을 위해 KVex.StationLog 모듈을 사용한다. Station Log가 생성되는 위치와 파일이름의 규칙은 다음과 같다.

- 디렉토리 : /home/kvg/KVGCS/Log/VLBI
- 파일이름 규칙 : <관측코드><3 character Station Code>.log
- 새 로그 파일에 대한 링크: KVGSI.log

StationLog의 각 정보는 시간 순서대로 기록되며 시각/키워드/값 양식을 갖는다. 시각은 DOYHHMMSS 양식으로 키워드의 값은 콤마로 구분된 값으로 표시된다.

예를 들어 r19049k 관측의 세종 Station 로그 파일에 기록되는 정보를 순차적으로 살펴보면 다음과 같다.

- 로그 파일 생성 코멘트 라인
Open log file for r19049kKSJ
- 관측 코드, 소프트웨어 버전, 세종 사이트 좌표
049105000/EXPER/ r19049k
049105000/KFS_VER/ 1.4.1, 1.6.1b
049105000/STATION/ V, SJ, KVGSEJONG, AZEL, -3110079.708, 4082066.845, 3775076.889, 0.0
- Mark6 기록기의 디스크팩 상태, 입력 스트림 정보
049105000/MK6_MSTAT/ 0, 1, 1, KVG00001/32000/4/8, 8, 8, 6388, 32000, closed, unprotected, sg, 2, 2, KVG00002/32000/4/8, 8, 8, 31967, 32000, open, ready, sg, 34, 3, KVG00003/32000/4/8, 8, 8, 1270, 32000, closed, unprotected, sg, 34, 4, KVG00004/32000/4/8, 8, 8, 2182, 32000, closed, unprotected, sg
049105000/MK6_STREAM/ 0, fila10g, vdif, 8224, 50, -1, eth5, 10.10.1.49, 46227, 0

- Mark6 기록기의 디스크 잔여 공간
049105000/MK6_RUNTIME/ 0, 2, 2048.0, 124871.09375, 31967, 32000
- 부경 초점 위치 (X, Y, Z, Tip Tilt)
049105000/PSUBR/ 1500.0, -1000.0, 14000.0, 0.0, 0.0
- Clock Offset (micro-seconds)
049105000/CLOCK/ -30.5
- 수신기 LO 주파수, 편광 (IF Num, LO/Pol)
049105026/LO/ 1, 21476.0
049105026/POLAR/ 1, L
- Video Converter의 설정 기록 (IF Num, VC Channel Num, LO)
049105026/BBC/ 1, 1, 256
049105026/BBC/ 1, 2, 256
049105026/BBC/ 1, 3, 256
049105026/BBC/ 1, 4, 256
049105026/BBC/ 1, 5, 256
049105026/BBC/ 1, 6, 256
049105026/BBC/ 1, 7, 256
049105026/BBC/ 1, 8, 256
049105026/BBC/ 1, 9, 256
049105026/BBC/ 1, 10, 256
049105026/BBC/ 1, 11, 256
049105026/BBC/ 1, 12, 256
049105026/BBC/ 1, 13, 256
049105026/BBC/ 1, 14, 256
049105026/BBC/ 1, 15, 256
049105026/BBC/ 1, 16, 256
- 기록되는 자료 양식 (VC Channel, 출력 채널, Sideband, Sampling Frequency, Sample Bit)
049105026/FORM/ 1, 1, LSB, 128000, 2
049105026/FORM/ 2, 2, LSB, 128000, 2
049105026/FORM/ 3, 3, LSB, 128000, 2
049105026/FORM/ 4, 4, LSB, 128000, 2
049105026/FORM/ 5, 5, LSB, 128000, 2
049105026/FORM/ 6, 6, LSB, 128000, 2
049105026/FORM/ 7, 7, LSB, 128000, 2
049105026/FORM/ 8, 8, LSB, 128000, 2
049105026/FORM/ 9, 9, LSB, 128000, 2
049105026/FORM/ 10, 10, LSB, 128000, 2

049105026/FORM/ 11, 11, LSB, 128000, 2
 049105026/FORM/ 12, 12, LSB, 128000, 2
 049105026/FORM/ 13, 13, LSB, 128000, 2
 049105026/FORM/ 14, 14, LSB, 128000, 2
 049105026/FORM/ 15, 15, LSB, 128000, 2
 049105026/FORM/ 16, 16, LSB, 128000, 2

- 스캔 정보가 코멘트 라인으로 기록

SCAN Start, Stop Time : 2019-02-18 11:16:10, 2019-02-18 11:17:00
 # No00001,GE01K_F

- 스캔의 천체 정보 (이름,

049105026/SOURCE/ 0336-019, \$, 03, 39, 30.937785, -01, 46, 35.80391, 2000.0, 0.0, 0.0

- Sky Dip 측정 결과 (IF Num, Tau0, , Tatm, ,)

049105407/SECZ/ 01, 0.148280963445, 0, 266.8, 0.0, 0.0
 049105407/SECZ/ 02, 0.141501387788, 0, 266.8, 0.0, 0.0

- 천체 도달 및 Tracking 기상 정보

049105427/ONSOURCE/

- 기상 정보 (온도(C), 기압, 습도)

049105427/WX/ 3.8, 999.5, 41.2

- 기록 시작 및 Total Power 자료의 스캔 번호

049111609/MK6_REC/ ON
 049111609/ST/
 049111609/TPREC_ON/ 131041

- 시스템 온도 측정 값 (IF Num, 고도, 수신기 잡음온도, 시스템 온도)

049111617/TSYS/ 01, 42.9922700395, , 146.261273888
 049111617/TSYS/ 02, 42.9922700395, , 257.480530979

- 안테나 지향 오프 셋 값 (방위각, 고도 방향 [arcsec])

049111617/PTOFF/ 0.0, 270.0

- 스캔 종료 및 기록 종료

049111659/MK6_REC/ OFF
 049111659/ET/
 049111659/TPREC_OFF/

- 기록된 스캔의 정보

050110450/MK6_SCAN/ 0, 2, 3482, r19049k_KVGSJ_050110409, 1, -, OK, vdif,
2019y050d11h04m10s, 39.39, 10.122, 2.056, 0

- Mark6 기록 디스크 남은 공간 상태

050110821/MK6_RUNTIME/ 0, 2, 2048.0, 94792.96875, 24267, 32000

- Station Log의 자동 전송

Transferring log file to AOC
End File Transfer
End Experiment : r19049k

Mark6 관련된 정보는 Mark6의 명령에 대한 응답이 기록된다. 이 부분은 Mark6 Command Manual에 기술되어 있다.

주 의 사 항

1. 본 보고서는 국토교통부 국토지리정보원의 수탁을 받아 한국천문연구원에서 수행한 연구보고서입니다.
2. 본 내용을 대외적으로 게재, 인용할 때에는 반드시 국토교통부 국토지리정보원의 사전 허락을 받기 바라며, 무단 복제는 절대 금합니다.

인쇄 · 2019년 12월

발행 · 2019년 12월

발행자 · 사공호상

발행처: 국토교통부 국토지리정보원

주소 · 경기도 수원시 영통구 월드컵로 92(원천동)

전화 · 031-210-2700

FAX · 031-210-2644