

R2DBE System Setup

Pre-Checks

- Check that the 1pps is connected to all R2DBE units (frontside)
- Check that the IFs are connected to the Block Down Converter (BDC) with RCP and LCP as labeled on the backside
- Check that the 10MHz signal is connected to both clock synthesizer units
- Check that the 10MHz signal is connected to both BDC units
- Check that both IF input pins of all R2DBE units (frontside) are connected by cable
- Check that all R2DBE units are connected by ethernet (PPC net port) to the network hub box (backside)
- Check that all mark6 units are connected by ethernet cable to the the network hub box (backside)
- Check that all Mark6 units have eth3 connected to R2DBE CH0 and eth5 to R2DBE CH1.
- **(Optional)** Check that the noise generator unit is connected by ethernet to the network hub box (backside)
- **(Optional):** Insert a tone into the noise generator unit (backside)

Startup sequence

1. power on the control computer
2. power on the mark6 units
3. power on the clock synthesizer units
4. power on the BDCs
5. power on the R2DBEs
6. power on the noise generator **(optional)**

Verify unit startup

- log into recorder1 as user oper. If logging-in does not work the mark6 machine is probably stuck in booting. Connect a display and keyboard and check.

```
ssh recorder1 -l oper -X
```

- do the same for the other recorders
- log into the control computer as user oper. Verify that all the R2DBEs units have booted up by trying an ssh connection:

```
ssh r2dbe1 -l root
ssh r2dbe2 -l root
...
exit from r2dbe1, r2dbe2, ...
```

if one of the units does not boot consult the *Troubleshooting* section below.

Configuring the R2DBEs

- log-into the control computer as user oper:

```
ssh vlbicc -l oper
```

- stop the monitoring daemons

```
r2daemons_stop
```

- initialize all the R2DBE units by running:

```
start_backend.py
```

This will use the configuration stored in /etc/backends.conf

- start the monitoring daemons

```
r2daemons_start
```

Adjust IF power levels

The following should be done when the real VLBI IFs are connected and the receiver is correctly tuned.

On the control computer as user oper do:

- adjust the 2bit sampling threshold:

```
alc.py r2dbe1
```

repeat for r2dbe2, r2dbe3 and r2dbe4

- adjust the BDC attenuation
alc.py will output the recommended attenuation settings. Use the front panel of the BDCs to set the attenuators accordingly. BDC1 is serving r2dbe1 and r2dbe2 whereas BDC2 is serving r2dbe3 and r2dbe4.
- start the monitoring tool:

```
r2dbe_monitor.py r2dbe1
```

Check the power levels. if too low or too high re-adjust the attenuator settings (see above)

Check the 2-bit sampling statistics. If not correct rerun alc.py

repeat for all R2DBEs

Troubleshooting

R2DBE unit does not boot

- Check that the R2DBE is connected by ethernet to the control computer (via the switch). If not connect and reset the R2DBE.
- Connect a USB cable from the R2DBE (backside "FTDI USB" port) to one of the USB ports on recorder1.
- log-into recorder1 as root
- run:

```
screen /dev/ttyUSB2 115200
```

- hit the "enter" key. The log-in prompt of the r2dbe unit should appear

- reset the R2DBE unit. In the screen window you should see the R2DBE booting. Inspect any messages that could indicate an error
- press ctrl-a to leave the screen session

R2DBE unit has booted but is not reachable from the mark6 machines

- very likely the R2DBE has been assigned the wrong IP-address by the *dnsmasq* service running on the control computer.
- log-into the control computer as user root
- run

```
/etc/init.d/dnsmasq restart
```

- reset the R2DBE unit