Mark5 control commands

Mark5 Control commands

The Mark5A protocol uses two TCP ports:

- m5data 2630/tcp # Mark5 data
- m5drive 2620/tcp # Mark5 control

Commands are sent to the **m5drive** port as newline terminated strings. The replies are newline terminated also.

Data is sent to the **m5data** port.

The following commands from the Mark5A command need to be sent to the Mark5 at the station:

- play_rate
- play
- play?
- net_protocol
- in2net
- mode
- status?
- mtu

A typical session would involve the following commands being sent:

→`mtu=9000` (if UDP)
`net_protocol=udp:8388608:131072:8`
`mode=mark4:32`
`play_rate=data:16`
`ipd=10` `in2net=connect:145.146.96.21`
`in2net=on`

Time passed

→`in2net=disconnect`

A more detailed description for the commands issued:

play_rate=data:<rate>

Used to set the output data rate of the Mark5 at the station.

Expected response:

[=!=]play_rate = 0 ;

play=off

Used to make the Mark5 stop playing if it was playing back data from disk.

net_protocol=<protocol>:<sockbuf size>:<workbuf size>

Used to set the network data-transport protocol. The <sockbuf size> argument is the socket send buffer size. You should probably use this value in a setsockopt(…, SOL_SOCKET, SO_SBDBUF, …) call on the socket used to send the data.

Expected response:

[=!=]net_protocol = 0 ;

mtu=<mtu size>

Set the mtu for UDP transfers

Expected response:

[=!=] !mtu = 0 : $mtu bytes;

in2net=connect:<ip>

Used to make the Mark5 at the station connect to the Mark5 at JIVE. The <ip> argument will be the IP address of the Mark5 at JIVE, which should at that point be ready to accept a connection from the Mark5 at the station. The connection should be made to the m5data port. No data will be sent yet.

Expected response:

[=!=]in2net = 0 ;

in2net=on

Used to make the Mark5 at the station start sending data.

Expected response:

[=!=]in2net = 1 ;

in2net=disconnect

Used to make the Mark5 at the station stop sending data and terminate the (tcp) connection.

Expected response:

[=!=]in2net = 0 ;

in2net=off

??

Expected response:

[=!=]in2net = 0 ;

mode=<data mode>:<data submode>

Used to set the playback mode of the Mark5 at the station.

Expected response:

[=!=]mode = 0 ;

status?

Used to query the status of the (remote) Mark5.

Expected response:

[=!=]status? 0 : 0x00000001 ; (when not sending data)
[=!=]status? 0 : 0x00010001 ; (when sending data)

play?

Request play state (disk only??)

Expected response:

[=!=]status? 0 : 0x00000001 ; (when not sending data)
[=!=]status? 0 : 0x00010001 ; (when sending data)

Setting TVG test

To setup the Mark5 to sent thr TVG test vector, run:

mode=tvg:8 play_rate=data:4

This will send 8 tracks at 4 Mbps, ie a total of 32 Mbps.

Mark5 Control commands

The Mark5A protocol uses two TCP ports:

- m5data 2630/tcp # Mark5 data
- m5drive 2620/tcp # Mark5 control

Commands are sent to the **m5drive** port as newline terminated strings. The replies are newline terminated also.

Data is sent to the **m5data** port.

The following commands from the Mark5A command need to be sent to the Mark5 at the station:

- play_rate
- play
- play?
- net_protocol
- in2net
- mode
- status?
- mtu

A typical session would involve the following commands being sent:

→`mtu=9000` (if UDP)
`net_protocol=udp:8388608:131072:8`
`mode=mark4:32`
`play_rate=data:16`
`ipd=10` `in2net=connect:145.146.96.21`
`in2net=on`

Time passed

→`in2net=disconnect`

A more detailed description for the commands issued:

play_rate=data:<rate>

Used to set the output data rate of the Mark5 at the station.

Expected response:

[=!=]play_rate = 0 ;

play=off

Used to make the Mark5 stop playing if it was playing back data from disk.

net_protocol=<protocol>:<sockbuf size>:<workbuf size>

Used to set the network data-transport protocol. The <sockbuf size> argument is the socket send buffer size. You should probably use this value in a setsockopt(..., SOL_SOCKET, SO_SBDBUF, ...) call on the socket used to send the data.

Expected response:

[=!=]net_protocol = 0 ;

mtu=<mtu size>

Set the mtu for UDP transfers

Expected response:

[=!=] !mtu = 0 : $mtu bytes;

in2net=connect:<ip>

Used to make the Mark5 at the station connect to the Mark5 at JIVE. The <ip> argument will be the IP address of the Mark5 at JIVE, which should at that point be ready to accept a connection from the Mark5 at the station. The connection should be made to the m5data port. No data will be sent yet.

Expected response:

[=!=]in2net = 0 ;

in2net=on

Used to make the Mark5 at the station start sending data.

Expected response:

[=!=]in2net = 1 ;

in2net=disconnect

Used to make the Mark5 at the station stop sending data and terminate the (tcp) connection.

Expected response:

[=!=]in2net = 0 ;

in2net=off

??

Expected response:

[=!=]in2net = 0 ;

mode=<data mode>:<data submode>

Used to set the playback mode of the Mark5 at the station.

Expected response:

[=!=]mode = 0 ;

status?

Used to query the status of the (remote) Mark5.

Expected response:

[=!=]status? 0 : 0x00000001 ; (when not sending data)
[=!=]status? 0 : 0x00010001 ; (when sending data)

play?

Request play state (disk only??)

Expected response:

[=!=]status? 0 : 0x00000001 ; (when not sending data)
[=!=]status? 0 : 0x00010001 ; (when sending data)

Setting TVG test

To setup the Mark5 to sent thr TVG test vector, run:

mode=tvg:8 play_rate=data:4

This will send 8 tracks at 4 Mbps, ie a total of 32 Mbps.