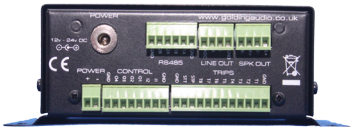
Golding Audio Ltd
8 Peartree Business Centre
Stanway
Colchester
Essex
CO3 0JN
T: +44 (0)1206 762462 F: +44 (0)1206 762633

DMS6000

MP3 player

Manual iss1.03 23-06-17



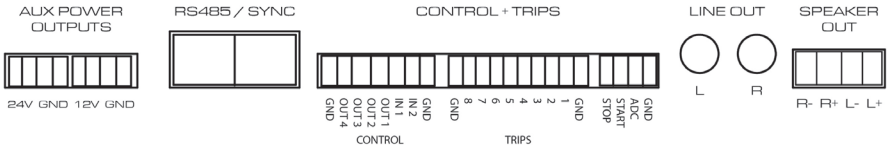
Features

- Cost effective
- Dual audio outputs
  - Highly efficient 2 x 25W Class D Amplifier
  - Unbalanced line level output
- 8 direct trip inputs (binary mode extends trips to 255)
- 4 fully controllable open drain outputs (30vdc, 2A max)
- Improved control functions
- Simple firmware upgrade
- Flexible ordering options
  - Board only
  - Compact case (DC powered)
  - 1U, 19" (mains powered)
  - 3U Subrack mounted (upto 8 boards per nest)

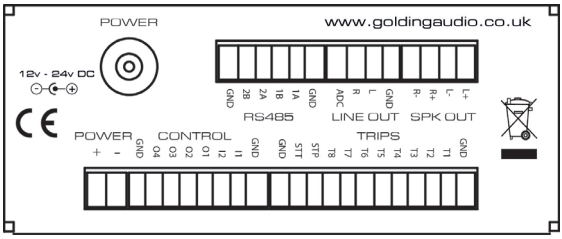
Index

Connections

1U Cased Unit rear connections



Compact Cased Unit / Nest unit connections



Trip Inputs 1 - 8 (T1 - T8)

Trip inputs can initiate playback of a stored message directly or start a command file which in turn can call any available messages.
The eight trip inputs can be configured as active HI or LOW via JP3. (default active LOW)
Trip inputs can be individually configured as normally open (N/O) or normally closed (N/C) inputs in the 'INIT.txt' file stored on the SD card. (default N/O)
Trip modes can be configured via the 'INIT.txt' file on the SD card
(5v logic, 10k internal pullup / down)

Stop Input (STP)

A LOW pulse on this line will stop and reset any running message or command file.
If the stop input is held low, it will prevent activation of any stored message or command file.
(5v active low, 10k pullup)

Start Input (STT)

When using binary trip inputs (255 messages) a LOW pulse on this line will latch in the binary code presented on the 8 trip input lines and initiate playback of the corresponding message or command file.
When using decimal trip inputs (8 messages) this input is inactive.
(5v active low, 10k pullup)

Control IN 1 - 2 (I1 - I2)

No Function. For future Use

Control OUT 1 - 4 (O1 - O4)

4 off, open drain (pull to ground) control outputs that can be configured via the output control files or command files on the SD card.
Any ouptut can be configured to behave as a RUN line. A run line is active (LOW) while an audio file is playing.
Be sure to use back-emf diodes if driving inductive loads such as a relay.
(30v / 2Amps maximum per output, 4A in total if all 4 used at once)

ADC input (Remote Volume Control / Timer Control)

This input can be used to externally control the volume, mute the audio completely or it can be used to set the value of one of the command file timers. See 'INIT.txt' file and 'Command.txt' file information.
ADC in requires a voltage between 0v and 10v to operate although you can use 12v but the last 10-12v won't make any further increase.

Volume control / Timer control.
Connect a 10K potentiometer wiper to the ADC input, the low end of the pot track to 0v and the high end of the track to 12v.
Or feed in 0-10v from an external source.

Mute control.
Connect a 10k resistor from ADC input to 12v and then ground the ADC input with a switch or relay to mute the audio.

RS485/Sync

Allows the DMS6000 to be controlled via RS485 serial commands with functions such as Play, Stop and Volume control.
This port is also used for the synchronisation of multiple DMS6000 units.
There are two RS485 busses.

Line out

1V rms into 600Ω unbalanced (+2.2dBu).

Speaker output (SPK OUT)

Compact Case /board only	12V supply, 8Ω load 8W + 8W. 4Ω load 15W + 15W
Compact Case / board only	24V supply, 8Ω load 13W + 13W, 4Ω load 27W + 27W
1U Unit,	8Ω load 13W + 13W, 4Ω load 27W + 27W

Aux Power Outputs (1U case Only)

Auxiliary 24Vdc and 12Vdc power outputs. Maximum load 1amp.

Power +/- (Compact case)

Auxiliary power output. Output voltage is the same as the Input voltage applied to the DC jack. Can also be used as power input connection.

Board Only Edge connector Pinout:

Row A		Row B	
1 *	Left line / speaker+	1 *	Left line / speaker+
2	Left Ground	2	Left Ground
3	Left speaker -	3	Left speaker –
4 *	Right line / speaker+	4 *	Right line / speaker+
5	Right Ground	5	Right Ground
6	Right speaker -	6	Right speaker –
7	Ground	7	Ground
8	Line out Left	8	Line out Right
9	ADC input	9	NC
10	Trip 1	10	Trip 2
11	Trip 3	11	Trip 4
12	Trip 5	12	Trip 6
13	Trip 7	13	Trip 8
14	Stop	14	Start
15	RS485-1B	15	RS485-1A
16	NC	16	NC
17	CTL in 2	17	CTL in 1
18	NC	18	CTL out 1
19	RS485-2A	19	NC
20	RS485-2B	20	NC
21	NC	21	Ground
22	NC	22	Ground
23	CTL out 2	23	CTL out 2
24	CTL out 3	24	CTL out 3
25	CTL out 4	25	CTL out 4
26	NC	26	NC
27	NC	27	NC
28	Ground	28	Ground
29	5V out	29	5V out
30	Ground	30	Ground
31	OV in	31	OV in
32	12-24V in	32	12-24V in

* Line or Speaker outputs are selected with jumpers JP1 & JP2 on board

Manual controls

Push button	Press to test play a file. The file number is defined in the 'INIT.txt' file.
Volume	Turn clockwise to increase the audio output from -64dB to 0dB
Bass	Turn clockwise to increase the low frequencies from 0 to +10dB The cutoff frequency is set in the 'INIT.txt' file.
Treble	Turn clockwise to increase the high frequencies from 0 to +15dB The cutoff frequency is set in the 'INIT.txt' file.
Balance / aux	Adjust the balance between left and right. Can also used for adjusting a variable timer in command file mode.

LED Indicators

Blue POWER LED	ON indicates power is applied and the board is running. Blinking indicates board is in low power standby mode. (future use)
Green PLAY LED	Audio playback in progress.
Yellow BUSY LED	Reading card for data.
Red STATUS LED	Brief flashes followed by pause indicates error codes. ON indicates Command file running or Synced if slave board. Flashing 50:50 indicates command file timer is running.

Error codes show on Status(Red) led and Run(Green) led while in boot loader.

1 No SD card
2 Flash update about to start
3 No firmware blown
4 No firmware file found
5 Bad firmware (.hex) file
6 Flash update failed

Error codes show on Status(Red) led only if in main program.

1 No SD card
2 No 'INIT.txt' file
3 No 'DATA' folder
4 ID tag err (ID tags are not allowed)
5 Invalid bit rate / sample rate

SD card.

The DMS6000 can use SD cards ranging in size from 1GB to 32GB.
The card must be formatted as FAT32.

Contents

The root of the card should have two files and a folder named DATA.

DATA
DMS6KREL V1_??.HEX
INIT DMS6K.TXT

The 'DMS6KREL V1_??.HEX' file is the firmware for the system and can be replaced with a new file to add more features as and when they become available. See 'Firmware update'

The 'INIT DMS6K.TXT' file is used to configure the operation of the DMS6000. See 'Configuration file'.

The 'DATA' folder will contain all of the audio files, control files and command files.

Audio file format

The DMS6000 currently supports .MP3 (Mpeg 1 Layer 3) files.
Useable Sample rates and bit rates are as follows.

Sample rates	32KHz, 44.1KHz and 48KHz.
Bit rates	32, 64, 96, 128, 160, 192, 224, 320 Kbps

Audio files

The mp3 Sound files must be named in the following manner to enable the DMS6000 to identify the files.

The first 3 digits in the file name assign the file to its trip input number.
Codes 001 to 008 are possible for direct trip inputs.
Codes 001 to 255 are possible if using binary trips or command files.

The next string of characters are used for your file name. Only the first 16 of these characters will be displayed on the 2x16 LCD display if fitted.

The last 4 characters must be .mp3 for a valid audio file name.

Example 1: **001 TEST MESSAGE.mp3**
Example 2: **156 This is a test.mp3**
Both the above are valid file names for messages 1 and 156 respectivley.

Configuration file 'INIT DMS6K.TXT'

The configuration file or 'INIT.txt' file defines the basic operation of the DMS6000.
It sets up the Trip input priority, tone control frequency and various other features.

The INIT file can be named however you like but the first 4 characters must be 'INIT' and the extension must be '.TXT'

The 'INIT.txt' file **must** be edited with a pure text editor such as 'Notepad' in Windows.

The INIT file contains the following data:

/Debounce	=5;	Trip Debounce time in 10mS increments
/StSpDeb	=5;	Start / Stop Debounce time in 10mS increments
/Invert	=00000000;	Normally closed bits, inverts trips '12345678'. Set to 0 or 1
/Oneshot	=00000000;	One shot bits for trips '12345678'. Set to 0 or 1
/Interrupt	=3;	Set the trip interrupt priority, see below.
/Binary	=0;	Input trips are binary encoded if 1. (Use START to latch code)
/Volume	=0;	value 0-64, 0=pot, 1-64=fixed volume.
/Bass	=0;	Value 0-15, value x 1dB. 0 to 15dB increase.
/BassFreq	=11;	Value 2-15, 20Hz to 150Hz (default is 110Hz)
/Treble	=0;	Value 0-7, value x 1.5dB. 0 to 10.5dB increase.
/TrebFreq	=5;	Value 2-15, 2KHz to 15KHz (default is 5KHz)
/Balance	=0;	Value 0-64, 0=pot, 1= full left, 32=centre, 64=full right.
/TestButt	=1;	Defines the file to play with the test button.
/AutoPlay	=0;	Auto-play file on power up if no other file playing
/ExtPot	=0;	Define function of external pot input (ADC)
/Ampli	=1;	Amplifiers disabled if 0, active if 1
/AmpSSM	=1;	Amplifier Spread Spectrum Mode (Set to '0' for headphones)
/RunChan	=1;	Defines which output is the run line 0-4. ('0' = not active)
/EndOut	=1;	Output file ends with audio if '1'
/Master	=0;	Define board as master in synced system.
/Slave	=0;	Define board as slave in synced system.
/AddrA	=00;	RS485 address, any value between 1 & 99
/AddrB	=00;	Global RS485 address, any value between 1 & 99
/RS485EN	=0;	Enable RS485 functions
/RSBAUD	=3;	RS485 baud rate (3 = 19200)

@ End of configuration data marker

Help:
Interrupt values:

0	No interrupts allowed
1	Lower trips interrupt higher ones playing
2	Any trip can interrupt any other one playing
3	Self interrupting allowed as well as any other interrupt

Detailed explanation of configuration file 'INIT.txt' contents

/Debounce = 5; **Debounce time in 10mS increments**
This parameter configures the time that the trip inputs have to be valid and stable before the DMS6000 will respond. A value of 5 means the trip has to valid for 50mS.
You may set any value from 1 to 500 (10mS to 5000mS)
This can normally be left at 5 but if a particularly electrically noisy environment is causing false triggers, then raise this value to try and fix the problem or if the trigger pulse is very short such as from an Infra Red sensor, try reducing this value.

/StSpDeb = 5; **Start / Stop Debounce time in 10mS increments**
This parameter configures the time that the START and STOP inputs have to be valid and stable before the DMS6000 will respond. A value of 5 means the input has to valid for 50mS.
You may set any value from 1 to 500 (10mS to 5000mS)
This can normally be left at 5 but if a particularly electrically noisy environment is causing false triggers, then raise this value to try and fix the problem or if the input pulse is very short, try reducing this value.

/Invert = 00000000; **Invert trips**
This parameter allows you to invert the operation of any or all of the trip inputs so that the trip is active when the trip line is open.

For example:
/Invert = 10000000; Trip 1 is inverted. File 1 plays when Trip 1 is inactive.

The above example could be used when the DMS6000 is connected to a standard PIR sensor which has a normally closed contact output that opens when the PIR detects movement.

The numbers following the '='sign can only be '1' or '0'
Trip 1 is the first digit, Trip 2 is the second digit etc
So for example if you wish to invert trips 3 & 4, you would set the command as follows:
/Invert = 00110000; Trips 3 & 4 inverted

/Oneshot = 00000000; **Edge triggered trips**
This parameter configures the trip inputs individually to allow one shot or edge triggering.
One shot triggering is used to prevent a file running continuously even if the trip remains active at the end of the file. (file could be MP3 or command file)

To re-trigger the file, the trip must be released and then re-applied after the file has finished.
This would normally be used when triggering from a contact that remains closed for longer than the sound track or command file such as with certain PIR sensors and timer switches.

For example:
/Oneshot =10000000; Trip 1 is one shot enabled.

The numbers following the '='sign can only be '1' or '0'
Trip 1 is the first digit, Trip 2 is the second digit etc.
So for example you wish to one shot trips 1,4 & 7, you would set the parameter as follows:
/Oneshot = 10010010; Trips 1,4 & 7 one shot enabled

/Interrupt = 0; **Trip interrupt mode for hardware trip inputs**
This parameter configures how the DMS6000 responds when another trip comes in while playing.

The four options are:
/Interrupt = 0; No interrupt
While a file is playing, all trips are ignored.

/Interrupt = 1; Priority interrupt
While a file is playing, trips of a lower value will be accepted and the current file will be cancelled.

/Interrupt = 2; Any other interrupt
While a file is playing, any other trip will be accepted and the current file will be cancelled.
If two trip inputs are active at the same time, the lowest one will take priority.

/Interrupt = 3; Any and self interrupt
While a file is playing, any other trip will be accepted and the current file will be cancelled.
If the same trip is released and then re-applied, the file will be restarted from the beginning.

If two trips are active at the same time, the lowest one will take priority.
More options may become available in the future.

/Binary = 0; **Binary coded trips**
Setting this parameter to '1' will allow access to 255 audio or command files by applying the appropriate binary code number on to the trip inputs and then applying the active low START signal.

/binary =0; up to eight files can be accessed with the 8 trip lines
/binary =1; up to 255 files can be accessed using binary codes

Binary equivalent value for each trip input.

Trip 1 = 1,	Trip 2 = 2,	Trip 3 = 4,	Trip 4 = 8,
Trip 5 = 16,	Trip 6 = 32,	Trip 7 = 64,	Trip 8 = 128.

Example:
To play file '043 name.mp3' you would apply trips 1, 2, 4, & 6 then apply an active LOW to the start Pin.
This equates to 1+2+8+32 = 43

The START line can be held low if the trips for the binary code are applied at the same time, ie from a logic board or diode matrix.

/Volume = 0; **Pre-set Volume**

This parameter allows the output level to be pre-set thus preventing operators from adjusting the front panel volume control.
If this command is not present or is set to zero, the front volume control will operate as normal.

The range for this item is 0-64 in 1dB steps where 64 = 0dB.

Formula for arriving at value for data byte.
value = 64 - dB

Examples:

/Volume = 64;	0dB	Max Vol
/Volume = 58;	-6dB	
/Volume = 24;	-40db	
/Volume = 1;	-63dB	MUTE
/Volume = 0;	Front control active	

/ExtPot = 0; **Define external analogue input function ADC**

This parameter defines the operation of the ADC input. It can be used to control the volume or it can be used to set one of the command file timers.

If using the ADC input for volume or mute control you should set the parameter to '1'

/ExtPot =0;	This disables the external ADC input.
/ExtPot =1;	This enables the external ADC input to control the volume.
/ExtPot =2;	This enables the external ADC input to control Timer 1.
/ExtPot =3;	This enables the external ADC input to control Timer 2.

/Bass = 0; **Pre-set Bass boost 0 - 15dB**

This parameter allows the BASS boost level to be pre-set thus preventing operators from adjusting the front panel Bass control.
If this parameter is not used or is set to zero, the front bass control will operate as normal.

The range for this item is 0-16 in 1dB steps

Examples:

/Bass = 0;	Using front panel Bass control pot, 0-15dB boost.
/Bass = 1;	Bass boost set to 0, no boost.
/Bass = 2;	Bass boost set to +1dB boost.
/Bass = 16;	Bass boost set to +15dB boost

/BassFreq =11; **Bass boost frequency cut-off**

This parameter defines the cut-off frequency of the Bass boost function.

The range for this item is 2 to 15 which equates to 20Hz to 150Hz
The default is 11 which equates to 110Hz.
If the parameter is not listed or is set out of range then the default of 11 will be used.

/Treble = 0; Pre-set Treble boost 0 - 10.5dB

This parameter allows the TREBLE boost level to be pre-set thus preventing operators from adjusting the front panel Treble control.
If this parameter is not listed or is set to zero, the front Treble control will operate as normal.

The range for this item is 0 - 8 in 1.5dB steps

Examples:

/Treble = 0;	Using front panel Treble control pot, 0 - 10.5dB boost.
/Treble = 1;	Treble boost set to 0, no boost.
/Treble = 2;	Treble boost set to +1.5dB boost.
/Treble = 8;	Treble boost set to +10.5dB boost

/TrebFreq = 5; Treble frequency cut-off

This parameter defines the cut-off frequency of the Treble boost function.

The range for this item is 2 to 15 which equates to 2KHz to 15KHz
The default is 5 which equates to 5KHz.
If the parameter is not listed or is set out of range then the default of 5 will be used.

/Balance = 0; Pre-set balance

This parameter can be used to pre-set the balance between left and right channels if you wish to re-purpose the balance control as a timer function.

The range for this parameter is 0 - 64.

Examples:

/Balance = 0;	Front panel control used to set balance.
/Balance = 1;	Pre-set balance to full left. (Right channel muted)
/Balance = 32;	Both left and right channels are equal level.
/Balance = 64;	Pre-set balance to full right. (Left channel muted)

If the parameter is not listed or is set to 0 then the front panel control will operate as normal.

/TestButt; Defines which file to play when pressed

This parameter defines which MP3 or command file to run when the test button is pressed.
The range for this parameter is 0-255 where 0 disables the button, 1 will play file 001, and 255 will play file 255.

Example:
/TestButt = 4;

When the test button is pressed, file numer 004 will be played. The file could be a 'Command' file or 'MP3' file

Default value is 1 if not listed.

/AutoPlay = 0; **Play this file if no other trip active.**

This file number will play if no other trips are active. It can be used to play a file on power up without having to activate any trip inputs.

Interrupt priority encoding of trips still applies.

Example 1:
/Interrupt = 1;
/AutoPlay = 4;

- Power on, no trips active.
- Plays track 4 due to auto play parameter.
- Trip 5 is now activated, nothing happens initially because track 4 has higher priority due to Interrupt=1;
- At the end of track 4 (loop point), track 5 will then run if trip 5 is still active.
- Trip 3 activated, track 4 stops and track 3 plays because track 3 has higher priority due to Interrupt=1;
- When track 3 ends and assuming trip 3 is released, track 4 will run again.

Example 2:
/Interrupt = 2;
/AutoPlay = 4;

- Playing track 4 due to auto play parameter.
- Any trip activated, track 4 stops and new track plays.
- Track 4 plays again when previous track ends and trip is released.

Notes:
If STOP pin is activated, all playback will stop and if a trip line is active upon release of the STOP line then the active trip will take precedence over the AutoPlay file no matter what the Interrupt paramter is set to.

When the tripped file ends and the trip is released, the AutoPlay file will play.

The trip lines will also have priority over AutoPlay on power up if a trip is active no matter what Interrupt is set to.

/Ampli = 1; **Amplifiers enable.**

This parameter can be used to disable the on-board power amplifiers if they are not required when only using line level outputs.

Examples:

/Ampli	= 0;	On board power amplifiers are disabled
/Ampli	= 1;	On board power amplifiers are enabled (Default)

/AmpSSM = 1; **Amplifier Spread Spectrum Mode.**

This parameter can be used to disable Spread Spectrum Mode on the Class D Power amplifiers on board.

SSM is normally enabled as it reduces electrical noise that could be radiated from the speaker cables.

You would need to disable SSM if you were to connect headphones to the amplifier.

If using headphones with a common ground (normally the case) then use only the positive amplifer outputs via series capacitors (100uF) and connect the headphoine common to ground.

The capacitor +ve should connect to the DMS6000 +ve speaker outputs.

This is because the amplifier complimentary outputs will produce distortion if not used as a pair when SSM is active.

/AmpSSM	=1;	Spread spectrum mode is active. Default
/AmpSSM	=0;	Spread spectrum mode is disabled.

/RunChan = 1; **Define RUN output.**

This parameter can be used to define which of the CTL outputs will be used as the 'RUN' line.

A run line goes active (low) while an MP3 file is playing.

Examples:

/RunChan = 0;	RUN signal not generated.
/RunChan = 1;	CTL1 output will act as the RUN line
/RunChan = 4;	CTL4 output will act as the RUN line

Note:

The Command file or Output file will over-ride the run output if being used in those files.

/EndOut = 1; **Output file ends with audio.**

This parameter can be used to force a running 'Output.txt' file to end when the audio file finishes.

/EndOut = 0;	A running Output file can continue beyond the MP3 if longer
/EndOut = 1;	A running Output file will end when the MP3 ends

Syncing and RS485 control features. See section to rear of manual

/Master =0;	Define board as master in Synced system
/Slave =0;	Define board as Slave in synced system
/AddrA =0;	RS485-1 address, any value between 1 & 99
/AddrB =0;	Global RS485-1 address, any value between 1 & 99
/RS485EN =0;	Enable RS485-1 functions
/RSBAUD =3;	RS485-1 baud rate (3 = 19200)

Output Control Files

An output control file can be written to drive the four Control output Mosfets, CTL out 1-4.
These outputs can drive relays, solenoids, LED's or other similar loads.
Each output can be programmed with multiple ON and OFF periods with a precision of 10mS up to 1 second per step.
Each channel can have up to 500 program steps.
Output Control files reside in the 'DATA' folder alongside the .MP3 files

- An output control file must be associated with an audio file but the audio file can be shorter or longer than the duration of the output control file.
- The output control file (if exists) will be loaded and started at the same time that the corresponding audio file is played.
- A running output control file will be cancelled by another valid trip being received.
- The STOP pin will cancel an output control file.
- For looping a control file, ensure that the control file is close to the audio file length as the output control file will loop at the loop point of the audio file, not the control file end point.

The output control file will be named as follows.
Q001 file name.txt

Q	- denotes a control file
001	- assigns the trip number that activates it
file name	- can be any text you wish (valid chrs only)
.txt	- the file extension for a text file

Output Control file syntax

Use a pure text editor such as Notepad for creating the Output control files.
Text can be upper or lower case or a mixture of both.
Each command must start with a greater than symbol '>' and be the first character of a new line.
Tabs and white space are allowed for clarity.
You may write comments after each command but the comment must be prefixed with a semicolon ';'

These are all valid entries:

```
>On 1;
>On1
>on=1
>On      1;      comment
```

Output Control file commands

>CHAN Defines the CTL output channel that the following program list will control.

>TICK Timer pre-scaler in 10mS increments, range 1 to 6000
10 = 0.1 second steps
100 = 1sec steps
6000 = 1 minute steps
Can only be set once per channel program. (To be set on first line)

>ON On time, range 1-250 (x TICK)

>OFF Off time, range 1-250 (x TICK)

>MRK Return point for repeat command. Can be used more than once in a program list.

>RPT Repeat command, range 0 to 250 repeats.
0 causes infinte repeats.
1 means the sequence will repeat once (commands run through then repeat once again)
2 means the sequence will repeat twice (runs through 3 times)
Repeat can be used more than once in a program but can't be nested.

>END End of channel program.

Note:
It takes about 100mS to read an 18KB file which equates to roughly 250 steps per channel.
This can affect the trip to audio out delay time.
The file can have up to 500 steps per channel.
Smaller files will have virtually no impact on trip delay.

The length of time that an Output Control file will run for is defined by the Channel with the longest run time. The shorter channels will be left in their last set state.

Example Output Control file:

```
>CHAN    1       ; CTL output 1 (the following items relate to control output channel 1)
>Tick    100     ; 1 second steps (100 x 10mS)
>on      1       ; Channel 1 on for 1 second
>off     4       ; Off for 4 seconds
>on      2       ; On for 2
>off     3       ; Off for 3
>end             ; End of this channel data

>CHAN2           ; CTL output 2
>Tick    10      ; 100mS steps
>mrk             ; Repeat marker
>ON      1       ; Channel 2 on for 100ms
>off     1       ; Off for 100ms
>On      2       ; On for 200ms
>Off     2       ; Off for 200ms
>rpt     9       ; Repeat the above sequence a further 9 times (10 runs)
>end             ; End of this channel data

>CHAN3           ; Ctl output 3
>Tick    1000    ;10 Second steps
>mrk             ; Repeat marker
>ON      5       ; On for 50s
>off     1       ; Off for 10s
>On      2       ; On for 20s
>Off     2       ; Off for 20s
>rpt     2       ; Repeat the above sequence twice
>end             ; End of this channel data
```

Channel 4 was not defined in this example so CTL out 4 will remain inactive for this trip number.

Command Files

Command files can be used to configure timers, sequencers and many other interactive functions to trigger the MP3 files.
Command files reside in the 'DATA' folder alongside the .MP3 files and Output Control files.

The command file will be named as follows:
C001 file name.txt

C – denotes a COMMAND file
001 – assigns the trip number that activates it
File name - can be any text you wish (valid chrs only)
.txt – The file extension for a text file

Command files can be created to add interactivity and special operating features to the DMS6000.

Command file syntax

Use a pure text editor such as Notepad for creating the command files.
Text can be upper or lower case or a mixture of both.
Each command must start with a greater than symbol '>'.
Each command will be followed by an '=' sign and then the value for the command.
The value will either be a decimal value or a bit field.
Tabs and white space are allowed for clarity.
You may write comments after each command but the comment must be prefixed with a semicolon ';'
The end of the command file should be marked with the '@' symbol.
You may type anything after the '@' such as description of the file operation.
The '>' and the '@' must be the first character of the new line.

These are all valid entries:

```
>Play       =1;
>play=1;
>PLAY       =          1;          comments
>PLay       = 001;                 comments
@
```

These are not valid entries:

Command			**Problem**
Play =1;			Missing '>'
>Play 1;			Missing '='
>Play	=1	comments	Missing ';' before comment
>Play=1;		comments	The '>' is not the first character of the line
@			The '@' is not the first character of the line

List of available commands.

Command	Value	Description	
PLAY	1-255	Play mp3	
PLAYL	0-255	Play mp3 looped	
STOP	000-111	Stop mp3, output or cmd file	
VOLUME	0-64	Preset the volume	
VOLDN	0-64	Volume decrement	
VOLUP	0-64	Volume increment	
VOLMIN	0-64	Minimum volume for VOLDN	
VOLMAX	0-64	Maximum Volume for VOLUP	
EXTPOT	0-3	EXTPOT parameter over-ride	
TIMER1	0-255	Timer 1 setting	
TIMER2	0-255	Timer 2 setting	
TIMOPT1	00000-11111	Timer 1 option bits	
TIMOPT2	00000-11111	Timer 2 option bits	
TIMRNG1	10-240	Timer 1 range when using variable input	
TIMRNG2	10-240	Timer 2 range when using variable input	
INTERRUPT	0-3	Interrupt parameter over-ride	
OUTSET	0000-1111	Explicit setting of all CTL outputs at once	
OUTON	0000-1111	Set specific CTL outputs to ON	
OUTOFF	0000-1111	Set specific CTL outputs to OFF	
RNDSTT	1-255	Start of group for random MP3 playback	
RNDEND	1-255	End of group for random MP3 playback	
RAND	0-1	Play random file from group or reset previously played list	
SEQSTT	1-255	Start of group for MP3 sequencer	
SEQEND	1-255	End of group for MP3 sequencer	
SEQSEL	0-16	Select sequencer (17 in total)	
SEQU	0-3	Play message from sequence	
TRIP	0-255	Trip input to test for	
WAITOR	0000-1111	Wait for any selected	
WAITAND	0000-1111	Wait for all selected	
IF	0000-1111	Conditional branch (OR of selected bits)	
ELSE	na	Start of commands if condition is FALSE	
ENDIF	na	End of conditional commands	
GOTO	1-32	Go to marker number	
MARK	1-32	Marker number	
CYCLES	0-255	Go to loop count	
AUTOPLAY	0-255	Auto play parameter over-ride	
FADEOUT	0-64	Fade out and pause MP3	(Future use)
FADEIN	0-64	Un-pause and fade in MP3	(Future use)

Detailed description of commands.

All of the following commands will be described as follows:

Command	Range of value	Value type	Description
>PLAY	= 1-255;	(decimal value)	Play file
>STOP	= 000-111;	(Bit field)	Stop file

Bit fields explained.
The numbers following the '='sign can only be '0' or '1'
Bit 1 is the first digit, Bit 2 is the second digit etc.

Example:
In this example, bit 2 is set and bits 1 & 3 are cleared.
Bit field order 123
>STOP = 010;

Playback commands:

>PLAY = 1-255; (decimal value) Play file
Play an MP3 with a file number from 001 to 254
Any currently playing MP3 will be cancelled before the new one starts.
If same file is called again, nothing will change.

Example:
>PLAY = 0; Does nothing
>Play = 3; Plays '003 name.MP3'

NOTE:
Special case when using WAITAND or WAITOR or COND commands.
If >Play = 255; and >Trip = 255; then the received trip number will play the same number MP3 file as the trip number that is applied.

>PLAYL = 0-255; (decimal value) Play file looped
Play and loop an MP3 with a number from 001 to 254
0 will clear the loop flag so the MP3 stops at the end of the track.
It does not cause the playing file to stop immediately.

Example:
>PLAYL = 4; Plays '004 name.MP3' continuously looped until stopped.
>PLAYL = 0; The currently playing looped file will now stop at the end of the file.

NOTE:
Special case when using WAITAND or WAITOR or COND commands.
If >Play = 255; and >Trip = 255; then the received trip number will play the same number MP3 file as the trip number that is applied.

>STOP = 000-111; (Bit field) Stop a file running

Bit 1	Stops a playing MP3
Bit 2	Stops OUTS program
Bit 3	Stops Command file

Examples:

>STOP = 110;	Stop MP3 and Output program
>STOP = 001;	Stop command file (Quit current command file)

>AUTOPLAY = 0-255; (decimal value) Auto play parameter over-ride

This command can be used to change the state of the /AUTOPLAY parameter of the configuration file within a command file program. For example, this could be used to disable the /AUTOPLAY feature after a particular trip was activated.

Audio and Analogue control commands:

>VOLUME = 0-64; (decimal value) Pre-set the volume

This command allows the output level to be pre-set, over-riding the front volume control or external volume control.
If this command is set to zero, the front volume control or external control (if enabled) will operate as normal.

The range for this item is 0-64 in 1dB steps where 64 = 0dB.

Formula for arriving at value for data byte.
value = 64 - dB

Examples:

>Volume = 64;	0dB	Max Vol
>Volume = 58;	-6dB	
>Volume = 24;	-40db	
>Volume = 1;	-63dB	MUTE
>Volume = 0;	Front control active	

>VOLDN = 0-64; (decimal value) Volume decrement

This command will decrement the pre-set volume by the amount specified each time it is run.
You also need to set the >Volmin and >Volmax values (see below)

Basic example:

>Volmin = 30;	Set minimum
>Volmax = 50;	Set maximum
>Volume = 40;	Volume is pre-set to 40
>VOLDN = 4;	Volume is now 36
>VOLDN = 4;	Volume is now 32
>VOLDN = 4;	Volume is now 30 (low limit reached)

Note:
If the volume has not been pre-set, the volume will be set to the value specified by the command >VOLMIN and not decremented the first time.

>VOLUP = 0-64; (decimal value) Volume increment
This command will decrement the pre-set volume by the amount specified each time it is run.
You also need to set the >Volmin and >Volmax values (see below)

Basic example:

>Volmin = 30;	Set minimum
>Volmax = 50;	Set maximum
>Volume = 40;	Volume is pre-set to 40
>VOLUP = 4;	Volume is now 44
>VOLUP = 4;	Volume is now 48
>VOLUP = 4;	Volume is now 50 (High limit reached)

Note:
If the volume has not been pre-set, the volume will be set to 0 then incremented by the amount specified.

>VOLMIN = 0-64; (decimal value) Minimum volume for VOLDN
This command sets the minimum volume level when using the Decrement '>VOLDN' command.

This should be set before using the >VOLUP & >VOLDN commands.

>VOLMAX = 0-64; (decimal value) Maximum Volume for VOLUP
This command sets the maximum volume level when using the Increment '>VOLUP' command.

This should be set before using the >VOLUP & >VOLDN commands.

>EXTPOT = 0-3; (decimal value) EXTPOT parameter over-ride
This command can be used to change the function of the ADC input.
The ADC input can be used to control the volume or it can be used to set one of the command file timers.

If using the ADC input for volume or mute control you should set the command to '1'

Example:

>EXTPOT =0;	This disables the external ADC input.
>EXTPOT =1;	This enables the external ADC input to control the volume.
>EXTPOT =2;	This enables the external ADC input to control Timer 1.
>EXTPOT =3;	This enables the external ADC input to control Timer 2.

Note:
Using this command will over-ride the /EXTPOT setting in the configuration 'INIT.txt' file until the next power cycle.

Timers 1 & 2:

>TIMOPT1 = 00000-11111; (Bit field) Timer 1 option bits
>TIMOPT2 = 00000-11111; (Bit field) Timer 2 option bits
These commands setup the way the two timers will operate.
There are two option commands, one for each timer.
The timer options must be set before setting the timer to run.

Bit 1 Timer runs for 'n' seconds if '0' or 'n' minutes if '1'. ('n' being the timer value)
Bit 2 Timer runs once if '0' or continuously if '1'
Bit 3 if set to '1' show time on LCD display if timer running.
Bit 4 If set to '1' allow timer update if the pot is turned while the timer is counting.
Bit 5 Cancel timer if '1'. Bits 1 to 4 are ignored if this bit is '1'

>TIMER1 = 0-255; (decimal value) Timer 1 setting
>TIMER2 = 0-255; (decimal value) Timer 2 setting
This command sets the timer and starts it counting.
The timer can be used to trigger playback or delay playback. They can be used as a lockout timer to prevent new trips from triggering playback until the timer has finished. These are just a couple of examples, there are many other uses for the timers.

The value defines how long the timer will run for in conjunction with the '>TIMOPT?' option bits.

A value of '0' means the timer will use a user variable input such as the front panel 'aux' pot or the external ADC input.

A value of more than '0' will set the timer to the value specified. The timer can be in seconds or minutes depending on the option bits. (see >TIMOPTn above)

>TIMRNG1 = 10-240; (decimal value) Timer1 range if using variable input
>TIMRNG2 = 10-240; (decimal value) Timer2 range if using variable input
This command allows you to set the maximum that the timer can be set to when using a variable input such as the front panel 'aux' pot or the external ADC input.
Useful if you want the variable range to be say 60 sec/min instead of the default 240 sec/min.

You can set the maximum range of the pot to anything from 10 seconds to 240 seconds (or 10-240 minutes)

Example:

>EXTPOT = 2;	Use ADC input to control Timer1
>TIMRNG1=60;	Set maximum range to 60
>TIMOPT1=01100;	Timer1 is counting in seconds and displaying the count
>Timer1 =0;	Timer1 is using a variable input

External pot set fully clockwise equates to 60 seconds
External pot set halfway equates to 30 seconds

Output control:

>OUTSET =0000-1111; (Bit field) Explicit setting of all CTL outputs

This command will set all of the control outputs on or off simultaneously as defined by the bits.
A '0' will turn off the corresponding output (open)
A '1' will turn on the corresponding output (pull to ground)

Bit 1 Control out 1 (OUT 1 / O1)
Bit 2 Control out 2 (OUT 2 / O2)
Bit 3 Control out 3 (OUT 3 / O3)
Bit 4 Control out 4 (OUT 4 / O4)

Example:
>OUTSET = 1010; Outputs 1 & 3 are on and output 2 & 4 are off
>OUTSET = 1001; Output 1 remains on, output 3 is turned off and output 4 is turned on

>OUTON = 0000-1111; (Bit field) Set specific CTL outputs to ON

This command will turn individual outputs ON as defined by the bits.
A '0' will do nothing
A '1' will turn on the corresponding output (pull to ground)

Bit 1 Control out 1 (OUT 1 / O1)
Bit 2 Control out 2 (OUT 2 / O2)
Bit 3 Control out 3 (OUT 3 / O3)
Bit 4 Control out 4 (OUT 4 / O4)

Example:
>OUTSET = 0001; Outputs 1-3 are turned OFF and output 4 is turned ON.
>OUTON = 1100; Outputs 1 & 2 are turned on. Outputs 3 & 4 remain in their previous state.
>OUTON = 0010; Output 3 is turned on. Outputs 1, 2 & 4 remain in their previous state.
(all outputs would now be on)

>OUTOFF = 0000-1111; (Bit field) Set specific CTL outputs to OFF

This command will turn individual outputs OFF as defined by the bits.
A '0' will do nothing
A '1' will turn OFF the corresponding output (open)

Bit 1 Control out 1 (OUT 1 / O1)
Bit 2 Control out 2 (OUT 2 / O2)
Bit 3 Control out 3 (OUT 3 / O3)
Bit 4 Control out 4 (OUT 4 / O4)

Example:
>OUTSET = 1111; Outputs 1 - 4 are turned ON.
>OUTOFF = 1100; Outputs 1 & 2 are turned off. Outputs 3 & 4 remain in their previous state.
>OUTOFF = 0001; Output 4 is turned off. Outputs 1,2 & 3 remain in their previous state.
(Only output 3 would be left on)

Sequencer & random player:

Sequencer.
The sequencer can be used to play an MP3 from a group of MP3 files each time a trip is activated or a timer ends.
There are 17 sequencer groups available which means you can have different groups of MP3's for multiple trip inputs.

1. Use the '>SEQSEL' command to select the sequencer number first.

2. Then use the '>SEQSTT' and '>SEQEND' commands to set the range of files played within a sequencer.
   If the sequencer is outside the latest set limits, the sequencer will be reset to the >SEQSTT value.
   If using different sequencers in multiple command files, you should set the limits on entry to the command file as there are only one set of '>SEQSTT' and '>SEQEND' commands.

3. Finally use the >SEQU command to trigger the MP3 file playback from the sequencer group.

Note:
The last selected sequencer will remain selected until the next reboot but it is good practice to select the required sequencer within the command file.

>SEQSTT = 1-255; (decimal value) Start of group for sequencer
>SEQEND = 1-255; (decimal value) End of group for sequencer
These commands specify the group of MP3's to play within the selected sequencer.

>SEQSEL = 0-16; (decimal value) Select sequencer (17 in total)
This command specifies the sequencer to use, 17 sequencers in total. Normally use Sequencer one for trip1 and sequencer 2 for trip 2 etc.
If you don't select a sequencer, sequencer 0 will be used as default.

>SEQU = 0-2; (decimal value) Play MP3 from sequencer group
This command will play a file from the selected sequencer group and has 3 possible settings.

>SEQU = 0; reset the sequencer to the start of the group as defined by '>SEQSTT = ???'
>SEQU = 1; Play the next file from the sequencer group (forwards through group)
>SEQU = 2; Play the previous file from the sequencer group (backwards through group)

Random player.
The random playback command can be used in a similar way to the sequencer above but with this command the MP3's are picked at random from within a group.

To prevent the same MP3 from playing more than once, there is a 'files played list' associated with the random feature. This table has 255 flags which are marked as used whenever a MP3 is played randomly.
When all MP3's have been played in the group, the flags will be cleared and the MP3's will then be marked as available again.

Note:
It is possible for a single repeat when the table has been filled up and then emptied as it is possible that the first MP3 in the new random sequence may be the same as the last played MP3 of the previous selected group.

>RNDSTT = 1-255; (decimal value) Start of group for random playback
>RNDEND = 1-255; (decimal value) End of group for random playback
These commands inform the random player the range of values to search between.
These should be set before using the '>RAND' command.

>RAND = 0-1; (decimal value) Play random file or reset list

>RAND = 0; Resets the 'files played list' only (between >RNDSTT & >RNDEND)
>RAND = 1; Play random MP3 from group

Command flow control:

>TRIP = 0-255; (decimal value) Trip input to test for
This command defines the trip number used by the 'WAIT' and 'IF, ELSE' commands

>TRIP = 0; Checking for no trip (trip released)
>TRIP = 1; Checking for a trip value of 1 applied
>TRIP = 2; Checking for a trip value of 2 applied

>TRIP = 255; Checking for any trip value applied (value is saved for use by >PLAY=255;)
Note: Works with the 'WAIT' and 'IF / ELSE' commands only.

>INTERRUPT = 0-3; (decimal value) Interrupt parameter over-ride
This command can be used to over-ride the /Interrupt = ?; parameter of the 'INIT.txt' file.

The four options are:
>Interrupt = 0; No interrupt
>Interrupt = 1; Priority interrupt
>Interrupt = 2; Any other interrupt
>Interrupt = 3; Any and self interrupt
Note: See configuration file above for more detail.

Loop control.
The following three commands allow you to jump forward or backwards within the command file. They can be used to repeat certain commands many times or they can be used just to make the command file easier to follow.

>CYCLES = 0-255; (decimal value) Go to repeat count
Number of times to jump using GOTO command.

>CYCLES = 0;	GOTO will be ignored, next command after GOTO will be run
>CYCLES = 1;	GOTO will jump to MARK once.
>CYCLES = 2;	GOTO will jump to MARK twice.
>CYCLES = 254;	GOTO will jump to MARK 254 times.
>CYCLES = 255;	GOTO will jump to MARK forever, infinite loop.

>MARK = 1-32; (decimal value) Marker number
This defines where the >GOTO command will jump to.
It can be before or after the >GOTO command.
You can have multiple '>GOTO' and '>MARK' commands within the same file (up to 32)
You can also nest the commands.

>GOTO = 1-32; (decimal value) Go to marker number
This command will cause the program to jump to the position specified by >MARK with the same value as specified.
Note: The cycle count will be decremented after the jump.

Example:
>CYCLES = 2;
>MARK = 1;
Other commands
>GOTO = 1;
More commands

The 'other commands' will be run 3 times

Timer, MP3 end and trip test.
These three commands are used to test for the Timer reaching zero, for testing for the end of an MP3 file and for testing the trip input state.

>WAITOR = 0000-1111; (Bit field) Wait for any selected
The program will pause at this point until ANY of the selected states become true.

Bit 1 Test if Timer1 has reached zero.
Bit 2 Test if Timer2 has reached zero.
Bit 3 Test if MP3 has ended (can detect loop point if >PLAYL used in command file)
Bit 4 Test for trip number applied as defined by >TRIP = n; **

Example:
>WAITOR = 1010; The program will pause here until either the TIMER1 has timed out OR the playing MP3 has ended.

Note:
If only Bit 3 is set and no MP3 is playing, the program will pause indefinitely

>WAITAND = 0000-1111; (Bit field) Wait for all selected
The program will pause at this point until ALL of the selected states become true.

Bit 1 Test if Timer1 has reached zero.
Bit 2 Test if Timer2 has reached zero.
Bit 3 Test if MP3 has ended (can detect loop point if >PLAYL used in command file)
Bit 4 Test for trip number applied as defined by >TRIP = n; **

Example:
>WAITAND = 1010; The program will pause here until both the TIMER1 has timed out and the playing MP3 has ended.

Note:
If Bit 3 is set and no MP3 is playing, the program will pause indefinitely

**
You must set the value of '>TRIP=?;' before using either of the Wait commands.
If '>TRIP = 0;' then the bit4 state becomes true when NO trip is applied (or trip released).
If '>TRIP = 255;' the value of the applied trip is saved (for use by PLAY), and the bit4 state is set to true.

Conditional branch.
These three commands allow testing various conditions and to then run different commands based on whether the test was true or false.

>IF = 0000-1111; (Bit field) Conditional branch test
The program will continue to the next command IF any of the selected tests are true.
If none are true, the commands following the '>ELSE' will run instead.

Bit 1 Test if Timer1 has reached zero.
Bit 2 Test if Timer2 has reached zero.
Bit 3 Test if MP3 has ended (can detect loop point if >PLAYL used in command file)
Bit 4 Test for trip number applied as defined by >TRIP = n; **

**
You must set the value of '>TRIP=?;' before using either of the Wait commands.
If '>TRIP = 0;' then the bit4 state becomes true when NO trip is applied (or trip released).
If '>TRIP = 255;' the value of the applied trip is saved (for use by PLAY), and the bit4 state is set to true.

>ELSE ; (no value required) Start of commands if condition is FALSE
The >'ELSE' marks the start of the commands that will be run if the '>IF' test was false.

>ENDIF ; (no value required) End of conditional commands
The '>ENDIF' marks the end of the conditional branch. Normal program flow continues from here.

Example:
```
>IF      = 1000;   Test if Timer1 has timed out
>PLAY    = 1;      If Timer1 has timed out then play '001name.MP3'
>ELSE    ;
>PLAY    = 2;      If Timer1 has not timed out then play '002name.MP3'
>ENDIF   ;
'other commands'
@
```

Example command files:

Example 1. Background music with advert every 10 minutes.
This command file will play '002name.MP3' in a loop and every ten minutes '001name.MP3' will play.

```
>cycles    = 255;          repeat program loop forever
>timopt1   =11100;         minutes timing, looping timer, show time on display
>timer1    =10;            set to 10 minutes

>mark      =1;             marker for goto to return to
>IF        =1000;          T1 timed out ?
>play      =1;             Play '1' if timed out
>Waitor    = 0010;         wait for end of 001name.MP3 then jump to 'ENDIF'
>ELSE      ;
>playl     =2;             Play '2' looped if timer still running
>ENDIF     ;
>goto      =1;             repeat the commands
@
```

Example 2. Nested condition command.
This command file will play 001.MP3 every 10 minutes and 002.MP3 every 60 minutes.
It will also play 003.MP3 if trip 8 is applied at any time.

```
>cycles    = 255;          repeat forever
>timopt1   =11100;         minutes timing, looping timer, show time on display
>timer1    =10;            set Timer1 to 10 minutes
>timopt2   =11000;         minutes timing, looping timer
>timer2    =60;            set Timer2 to 60 minutes

>mark      =1;             marker for goto to return to

>trip      =8;             testing for trip 8
>IF        =1000;          Timer 1 timed out ?
>play      =1;                     play '1' if Timer1 timed out
>Waitor    =0010;                  wait for end of MP3 then jump to 'ENDIF'
>ELSE      ;
>IF        =0100;          check timer 2 if timer 1 not timed out
>play      =2;                     play '2' if Timer2 timed out
>Waitor    =0010;                  wait for end of MP3 then jump to 'ENDIF'
>ELSE      ;
>IF        =0001;          check fot trip 8 if timer 2 not timed out
>play      =3;                     play '3' if trip 8 is active
>Waitor    =0010;                  wait for end of MP3 then jump to 'ENDIF'
>ELSE      ;
>out       =1000;          Turn on output 1 if none of the above are true
>ENDIF     ;

>goto      =1;             Go to 'mark=1' above
>@
```

Example 3. Play a different MP3 every time Trip1 is activated

This command file will play an MP3 from a group of 20 files numbered from '011name .MP3' up to and including '030name.MP3'

```
>cycles     =255;      Infinite program loops
>SeqSel     =0;        Select the default sequencer
>SeqStt     =11;       Set start of group of MP3's to play
>SeqEnd     =30;       Set end of group of MP3's to play
>Trip       =1;        Looking for trip 1 to start playback

>mark       =1;        return point for goto
>WaitOR     = 0001;    Wait for active trip 1
>sequ       =1;        play an MP3 from the group
>waitor     = 0010;    wait for mp3 end
>Trip       = 0;       Looking for trip release (no trips)
>WaitOR     = 0001;    Wait for no trip
>goto       =1;        go and check trip 1 active again
@
```

Synchronisation and RS485 Control.

Synchronisation

Multiple DMS6000's may be connected together to form a multichannel audio playback system which can be either triggered or running in looped mode. This is very useful for surround sound type installations or themed dark rides where there are multiple scenes that need to be played in sync.

There will be one Master DMS6000 and one or more Slaves (upto 32 Slaves) in a synchronised system. The Master connects to the slaves via the RS485-2 bus in a daisy chain fashion which can be accomplished using standard CAT5 network leads plugged into the sockets on the back of the 1U cased units.
If using compact units or nest units then use twisted pair wiring between the Master and Slaves.

The Slaves are totally controlled by the Master via the RS485-2 bus, there is no need to connect trips or control RS485-1 bus to the Slave boards and in fact those inputs are ignored if the DMS is configured as a slave.

The sync function maintains the DMS6000's sync within about 30-40uS of each other which is more than acceptable to maintain phase of signals in a surround sound setup.

To start the synced playback, trigger the Master DMS using either the direct trip inputs or the RS485-1 bus.

Due to how the synchronisation works, there are various requirements that must be met for sync to work correctly.

Syncing requirements & notes.

- MP3 files must be 48KHz sample rate.
- Bit rate ideally should be 160Kbps constant bit rate.
- MP3 files must not have any ID tags. (ID3 tags etc)
- Ensure file sizes are identical whether looped or triggered.
- Do not connect trips on slaves, Master controls file number to play.
- Do not enable RS485 on Slaves, Master controls file number to play.
- Use SD cards between 2GB and 16GB.
- Format as FAT32 with 4096 Allocation unit size (Cluster size) before loading the files.

How it works.

Frame sync:

The Master is always sending a data string to the slaves every 24mS (48K sample rate packet rate), This string is received by all the slaves and is used to accurately synchronise the decoders on all boards in the sync group within a few 10's of microseconds.

Play:

The Master board receives a trigger command either from the RS485 port or from the trip lines.
The Master sends a Cue request to the Slave boards along with the track number and a zero timecode.
The Master and Slaves search for the same track number on their respective SD cards.
When cued up, (about 40-60mS) the Master will send the Play request to the slaves and then start playing on the next frame clock tick.

Re-sync:
Slave behind master.
The slave is continuously checking the received time code against its playing timecode and if the slave is behind the master, the slave will search forward to become ahead of the master and then wait a second or 2 for the master to catch up. When the timecodes are equal, the slave will continue playback.

Slave ahead of master.
The slave is continuously checking the received time code against its playing timecode and if the slave is ahead of the master by only a few frames, the slave will pause and wait for the master to catch up then continue playing.
If the slave is beyond a few seconds ahead, the slave will search from the beginning of the file until slightly ahead of the master and then wait a second or 2 for the master to catch up. When the timecodes are equal, the slave will continue playback.

RS485 Control

The DMS6000 can be controlled by other equipment such as a PC or control system by using the RS485-1 bus.
This allows multiple DMS6000's to be controlled using just one serial signal connected across all units.

Note:
Currently you may only control MP3 files, command files can't be triggered at present.

To be able to use RS485 control, you must first set some parameters whithin the 'INIT.txt' file.

INIT file settings:

/AddrA	**=1;**	RS485 address, any value between 1 & 99
/AddrB	**=99;**	Global RS485 address, any value between 1 & 99
/RS485EN	**=2;**	Enable RS485 functions
/RSBAUD	**=1;**	RS485 baud rate (1 = 9600)

/ADDRA = 1 - 99; Sets the Primary address
Set this to a value between 1 and 99
This should be set to address an individual DMS board, usually the board number in the rack

/ADDRB = 1 - 99; Sets the Secondary address
Set this to a value between 1 and 99
This should be set to address multiple DMS boards at once for functions such as STOP or MUTE

/RS485EN = 0 - 2; RS485 mode
Set this to the following values depending on requirements

/RS485EN =0;	Disable RS485 functionality
/RS485EN = 1;	Enable RS485 receive so the DMS can be controlled
/RS485EN = 2;	Enable RS485 transmit so the DMS can send response or status. See notes.

Notes: (/RS485EN = 2;)

1. Does not send if ADDRB is called.
2. Be aware that if multiple boards are on the same bus and 2 or more MP3's end simultaneously then it is possible that the return strings could clash and be corrupted.
3. If a string is sent at the same time that an MP3 ends iut is possible for the send string to clash with the returned string and become corrput.

/RSBAUD = 1 - 6; Set Baud rate for RS485
Baud rates:

1 = 9600	Factory default if /RSBAUD not defined in 'INIT' file
2 = 14400	
3 = 19200	
4 = 38400	
5 = 56000	
6 = 57600	

Commands

LIST	Returns the MP3 file names in 'DATA' folder
PLAY	Play MP3 once (1-255)
PLAL	Play MP3 looped (0-255) '0' will cancel the looping function
STOP	Stop
VOLM	Preset Volume 0-64
VOLD	Decrement the volume
VOLU	Increment the volume
BASS	Preset Bass 0-16
TREB	Preset Treble 0-8

RS485 Control protocol

>	Start character (0x3e)
9	Address byte 10's (Ascii)
9	Address byte 1's (Ascii)
:	Delimiter
PLAY	Command word
:	Delimiter
2	Track number 100's (Ascii)
5	Track number 10's (Ascii)
5	Track number 1's (Ascii)
<cr>	Carriage return (0x0d)

Note:

The address must be 2 characters and be between 00-99

The track number must be 3 characters and be between 000-255

System Responses:

<READY [cr] [lf]	Sent on power up, DMS is ready to receive commands
<ACK [cr] [lf]	Command accepted
<NACK [cr] [lf]	Command rejected, check syntax!
<ERR SD Read error[cr] [lf]	
<ERR No SD card[cr] [lf]	

RS485 commands in detail:

LIST

This command will send the file name and duration of each MP3 track found in the data folder starting at 001 and ending when no more files are found. (Gives up searching if >10 files not found consecutively)

Send:
>01:LIST:000[cr]

Receive:
<ACK [cr] [lf]
<001Yello - The Race.mp3 | 03:01 | [cr] [lf]
<002Hot In The City.mp3 | 03:27 | [cr] [lf]
<003Blockbuster.mp3 | 03:10 | [cr] [lf]
<004Legs.mp3 | 03:25 | [cr] [lf]
<005Are You Gonna Go My Way.mp3 | 03:28 | [cr] [lf]
<006The Passenger.mp3 | 03:28 | [cr] [lf]
<END OF LIST [cr] [lf]

Detail:

<	Start character
001Yello - The Race.mp3	Long file name (up to 39 chrs)
|	delimiter
03:01	Track duration in minutes & seconds
|	delimiter
[cr] [lf]	carriage return / line feed

Notes:
1. The filename will be truncated at 39 characters
2. The .MP3 extension will be sent
3. This command will cancel playback
4. The files must not have ID tags.

This will be shown instead of time if ID tag is present.
<008 Things.mp3 | Remove ID tag | [cr] [lf]

PLAY

This command will start an MP3 file playing.
It will cancel any currently playing file first then start the requested file number.
If the same file number is re-sent while playing, the file will re-start.

Example:
Play track 6 once on board 1

Send
>01:PLAY:006[cr]

Receive (if not currently playing)
<PLAY:006 The Passenger[cr] [lf]

OR receive if playing a track
<MP3 END [cr] [lf]
<PLAY:006 The Passenger [cr] [lf]

At the end of the file, the following will be sent:
<MP3 END [cr] [lf]

Note:
The filename will be truncated at 36 characters, the .MP3 extension will not be sent

PLAL

This command will start an MP3 file playing and keep playing in a continuous loop.
You may cancel the looping mode without stopping the playing file by sending the following:
>01:PLAL:000[cr] ;Clears the loop mode only

See PLAY above for syntax

STOP

This command will cancel any playing MP3 file.

Send
>01:STOP:001[cr]

Receive (if not currently playing)
<STOP:001 [cr] [lf]

OR receive (if playing)
<MP3 END [cr] [lf]
<STOP:001 [cr] [lf]

Note:
The data field should be set to 001 to cancel an MP3
Other values are not yet defined.

VOLM

This command will pre-set the volume level over-riding the front mounted volume control.
A value of 00 will re-enable the front volume control.
01 is minimum pre-set volume.
64 is maximum pre-set volume.

Send
>01:VOLM:020[cr]

Receive
<VOLM:020 [cr] [lf]

VOLD

This command will decrement the pre-set volume level by the amount defined in the data field.
Each time this command is sent, the volume will decrease by the amount specified.

Send
>01:VOLD:002[cr] ;reduce volume by 2 steps

Receive
<VOLM:018 [cr] [lf] ;confirm new level (was level 20 before cmd)

Note:
The volume will not decrement below the value set by /VOLMIN in the init.txt file.

VOLU

This command will increment the pre-set the volume level by the amount defined in the data field.
Each time this command is sent, the volume will increase by the amount specified.

Send
>01:VOLU:002[cr] ;Increase volume by 2 steps

Receive
<VOLM:022 [cr] [lf] ;confirm new level (was level 20 before cmd)

Note:
The volume will not increment above the value set by /VOLMAX in the init.txt file.

BASS

This command will pre-set the Bass level over-riding the front mounted Bass control.
A value of 00 will re-enable the front Bass control.

00	Using front panel Bass control pot, 0 - 15dB boost.
01	Bass boost set to 0, no boost.
02	Bass boost set to +1dB boost.
16	Bass boost set to +15dB boost

Send

>01:BASS:016[cr] ;Set Bass to +15dB boost

Receive

<BASS:016 [cr] [lf] ;Confirm

TREB

This command will pre-set the Treble level over-riding the front mounted Treble control.
A value of 00 will re-enable the front Treble control.

The range for this item is 0 - 8 in 1.5dB steps

00	Using front panel Treble control pot, 0 - 10.5dB boost.
01	Treble boost set to 0, no boost.
02	Treble boost set to +1.5dB boost.
08	Treble boost set to +10.5dB boost

Send

>01:TREB:008[cr] ;Set Treble to +10.5dB boost

Receive

<TREB:008 [cr] [lf] ;Confirm

Internal jumper settings

WARNING.
DISCONNECT MAINS SUPPLY BEFORE OPENING THE UNIT.

DISCHARGE ANY STATIC CHARGE BY TOUCHING SOMETHING EARTHED.

To access the jumpers, disconnect the power and carefully remove the lid by removing the lid screws.

JP1 - Select Line out or Speaker out on pin 1 of edge connector. (DMS3000 bus compatibility)

JP2 - Select Line out or Speaker out on pin 4 of edge connector. (DMS3000 bus compatibility)

JP3 - Select COM-5v for active low trip inputs (default) or select COM-GND for active high trip inputs

JP4 - Fit to terminate the RS485-1 bus if last board on bus. (Only required for long cable runs)

JP5 - Fit to terminate the RS485-2 bus if last board on bus. (Only required for long cable runs)

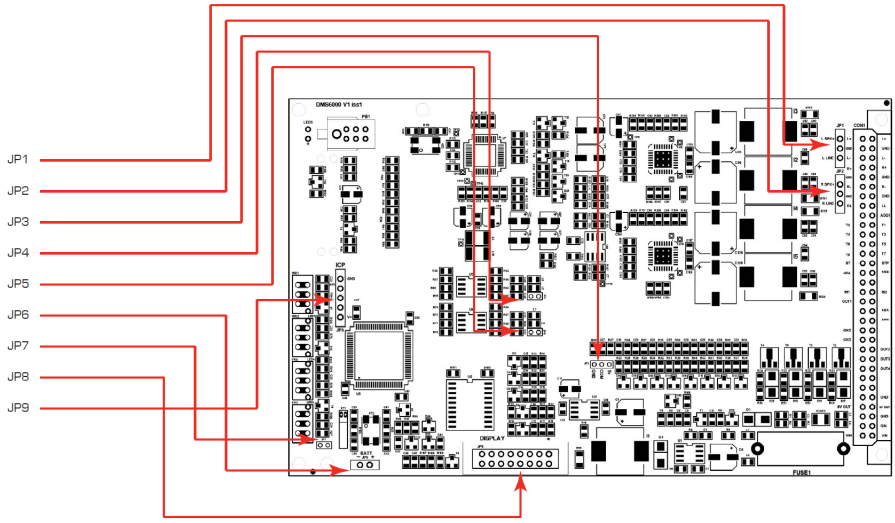
JP6 - Future use.

JP7 - For factory test only.

JP8 - Connect optional 2 x 16 LCD display.

JP9 - Factory use only.

JUMPER LOCATIONS



Specification:

Voltage input	10 – 28VDC MAX
Standby current	~100mA (~200mA with LCD)
Maximum current	~3A peak (driving 2 x 4 Ω speakers at 24v supply)
Line output	1V rms into 600Ω unbalanced (+2.2dBu)
Audio controls	Volume Treble boost Bass boost Balance
Speaker output power	Compact Case 12v supply, 8Ω load 8W + 8W Compact Case 12v supply, 4Ω load 15W + 15W Compact Case 24v supply, 8Ω load 13W + 13W Compact Case 24v supply, 4Ω load 27W + 27W 1U Case, 8Ω load 13W + 13W 1U Case, 4Ω load 13W + 13W
Memory	Standard SD card 1GB to 32GB (FAT32 format)
Audio file	MP3 (without ID tags)
Sample Rates	32KHz, 44.1KHz and 48KHz
Bit Rates	32, 64, 96, 128, 160, 192, 224, 320 Kbps
Trip inputs	8 Active low with 10k pullup to 5v **or** 8 Active high with 10k pull-down to gnd.
Trip modes	8 direct inputs Edge triggered Level triggered Normally open Normally closed Binary mode (255 possible codes) Priority interrupt Any interrupt No interrupt Self interrupt
Control outputs	4 x 2Amp open drain (pull to ground) 30v maximum. Any CTL out can be used as RUN output, active when playing
Control inputs:	
Start	Latches code in binary mode
Stop	Cancel any audio or control file
Control 1+2	Future use
ADC	Remote volume control (0-10v)
Remote control / Sync	RS485 serial control / Multiple board synchronisation

Troubleshooting

Please contact Golding Audio for technical support.