

Available Models

EM3038A (version 2.0)

Audio only, no relays.

EM3038A-R (version 2.0)

Audio + relays for simple control.

EM3038A-X (version 1.2)

Audio + relays for complex animation.

* Em3038A-X supports WAV files only.

Technical Specifications

Operation Mode

playback only

Sound File Format

Windows WAV (uncompressed PCM)

MP3 (ISO 11172-3)

Max. Number of Sound Files

Parallel / Direct Mode: 8

Parallel / Binary Mode: 128

Parallel / Sequential Mode: 511

Parallel / Round-Robin Mode: 8

Parallel / Script Mode: 511

Serial Control: 511

Memory Type

CompactFlash card, type I

Max. Memory Capacity

2 GB

Max. Recording Time

> 33 hours (for 128 kbps MP3)

Supply Voltage

10 ~ 32 VDC

Typical Standby Current

150 mA

Max. Audio Output

10W/channel (4 Ohm load)

Serial Interface

RS-232 / RS-485

Parallel Interface

8 inputs, CMOS logic level

Physical Dimensions

EM3038A

6.0" x 4.8" x 1.7"

EM3038A-R & EM3038A-X

6.0" x 7.5" x 1.7"

Firmware Version

2.0

Manual Parts

Part One: parallel interface

Part Two: serial port & application notes

* Part Two is available upon request.

Interface Descriptions

Power Input Terminals: VD & GD

Use a well regulated DC power supply to obtain the best sound quality. Connect the power supply's positive output to terminal VD, and the negative output to terminal GD.

Speaker Output Terminals: LF (left channel) & RT (right channel)

The speaker outputs are single ended (unbalanced). Depending on the ambient temperature, additional heat ventilation may be required to obtain maximum output. If the built-in power amplifier is not to be used, it can be disabled by moving the on-board jumper J3 to the OFF position.

Busy Output Terminal: BY

This output is manually controlled when the system runs in the Script mode. In all other modes this signal is automatically turned on during the playback of any file. The output is open collector and internally pulled up to +5V through a 10K resistor (R11). It drops to about 0.2V when activated, sinking current up to 100 mA. When controlling an external relay with this signal, be sure to remove resistor R11, and add a protection diode as shown in the diagram below.

System Reset Terminal: RS

Short terminal RS to the ground to reset the board. Minimum reset duration is 100 ms.

Parallel Port Terminals: T1 - T8

These inputs are internally pulled up to +5V through 10K resistors, and protected with 1K in-line resistors. The Parallel Port is enabled by default.

Power Light (PWR)

The power light is turned on when power is applied.

Volume Pot (VOL)

Turn the knob clockwise to increase the output volume. It affects both the speaker output and the line output.

Line Output (LINE): 1/8" Stereo Phone Jack

This jack provides single ended line output for external amplification.

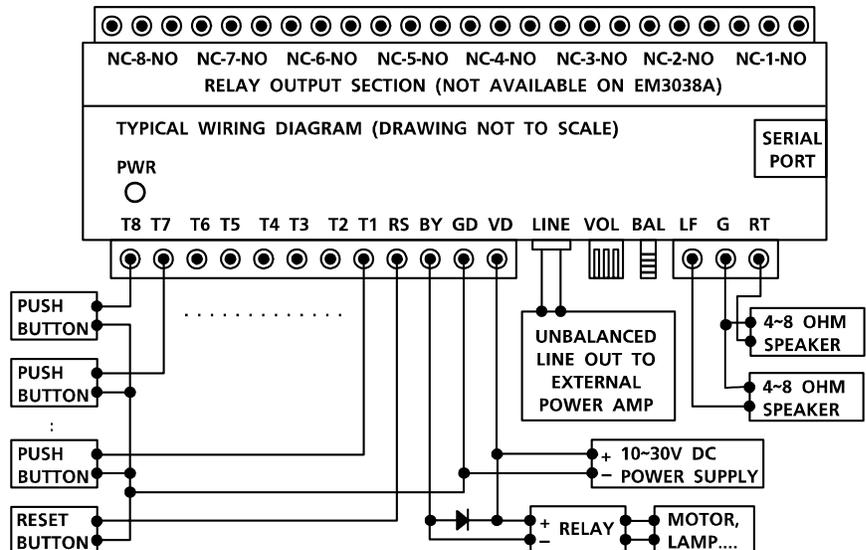
Serial Port: DB9 Female

The default serial mode is RS-232. To select RS-485, move the internal jumper J2 to the "485" setting. Note that the Serial Port is disabled by default.

Relay Output Terminals (EM3038A-R & EM3038A-X only)

Three terminals are provided for each relay: NC (Normally Closed), NO (Normally Open) and Common (the center terminal). The contact is rated at 12A/120VAC or 10A/24VDC.

Typical wiring diagram for push button applications



Parallel Trigger Modes

The Trigger Mode defines how the playback is to be triggered via the Parallel Interface which has 8 inputs labeled T1 through T8. All inputs are internally connected to +5V through 10K resistors. If left unconnected, an input is at +5V (logic "1"). These Trigger Modes do not apply when the system is controlled via the Serial Port.

Direct Trigger (default)

In this mode each input directly triggers a corresponding file: T1 = File 001, T2 = File 002,, T8 = File 008.

A trigger is valid when the input is shorted to the ground for at least 50 ms. The Direct Trigger is prioritized from T1 (the highest) to T8 (the lowest). However, it does not mean a higher priority input can interrupt a lower one. It only means that if multiple triggers are applied at the same time, the highest priority wins.

Binary Trigger

Use the Binary Trigger to trigger up to 128 different files ranging from 001 to 128. The Binary Trigger is often preferred when the system is controlled by an external controller because it requires less I/O ports in most cases.

To trigger a particular file, the first step is to signal the file number on T1 (LSB) ~ T7 (MSB). The signal must be in the binary format with +5V being logic "1", and 0V (the ground) being logic "0". For example, to signal File 007 (binary code "0000111"), T1 ~ T3 should be at +5V, and T4 ~ T7 should be at 0V. Note that, as a special case, the binary code for File 128 is "0000000".

The second and the last step is to short T8 to the ground for at least 50 ms while maintaining the signal on T1 ~ T7. Afterwards, signals on T1 ~ T7 don't matter any more.

Sequential Trigger

Use the Sequential Trigger to sequentially trigger up to 99 different files per input as defined below. However, the total number of files cannot exceed 512.

T1 triggers File 001 ~ 099

T2 triggers File 201 ~ 299

.....

T8 triggers File 801 ~ 899

Each trigger on the same input activates the next file in the sequence. The sequence automatically restarts when either the end of the sequence is reached or there is a break in the sequence. For example, if there are only three files on the flash card: 001, 002, and 004, the system will only sequence from 001 to 002. File 004 will never be played because File 003 is missing.

The Sequential Trigger is prioritized from T1 (the highest) to T8 (the lowest). However, it does not mean a higher priority input can interrupt a lower one. It only means that if multiple triggers are applied at the same time, the highest priority wins.

Round-Robin Trigger

This mode is very similar to the Direct Trigger mode except that the inputs are not prioritized. So if multiple inputs are tied to ground then their files will be played one after another, instead of just the highest priority one. Round-Robin mode can only be used in conjunction with Non-interruptible Playback and Script Playback.

Parallel Playback Modes

Non-interruptible Playback (default)

The file is played once per trigger. The playback is not interruptible except by the system reset. Looping is possible by applying a constant trigger on the input.

Interruptible Playback

The file is played once per trigger if not interrupted. Any input (including itself) can interrupt the playback later. Looping is possible by applying a constant trigger on the input.

Holdable Playback

The file is played for as long as the input is triggered, looping if necessary. It is not interruptible except by the system reset.

Script Playback

Please see the Script Playback Mode section for descriptions.

Parallel Trigger Polarity

Close Contact

The input is constantly triggered when it's at 0V (ground).

Open Contact

The input is constantly triggered when it's at 5V (open).

Make Contact

The input is triggered once when it's connected to the ground.

Break Contact

The input is triggered once when it's disconnected from the ground.

The Q Mode

The Q mode is disabled by default, and enabled by adding a letter Q as the fourth letter in the configuration file MODE.TXT. If the Q mode is enabled, the left channel output is inverted. For stereo files, this creates a virtual surround. For mono files, this creates a 4x more powerful differential output, but the balance control must be set at the middle and the speaker (8 Ohm recommended for this mode) must be connected to the LEFT and the RIGHT terminals.

Relay Control (EM3038A-R / EM3038A-X only)

The relays are controlled differently in different modes:

QWA mode (EM3038A-X only)

Relays are controlled in sync with audio playback. By using the Windows based QWAplay program, cue data are embedded into WAV files to generate QWA files. EM3038A-X automatically enters the QWA mode when playing QWA files.

Serial Interface mode

Relays are controlled by serial commands. Please see the Serial Commands section for descriptions.

All script modes: DS, BS, and RS

Relays are controlled by script commands. Please see the Script Playback Mode section for descriptions.

All other modes

A relay will turn on when the corresponding file is being played. Relay #1 corresponds to file #001, relay #2 corresponds to file #002, and so on. For example, relay #8 will be turned on when file #008 is being played.

Script Playback Mode

Instead of playing just a single file, Script Playback mode executes a script of commands for each trigger. Note that the Busy output will not turn on/off automatically in the Script mode. It must be manually turned on/off using the BN/BF commands.

Written in the configuration file using plain ASCII text, the script consists of multiple lines each containing the commands for a particular trigger in the following format:

```
?nnn=Command1,Command2...
```

Here "nnn" is the trigger number and "?" is one of the following:

N - Non-interruptible

Execution of this trigger is not interruptible.

I - Interruptible

Execution of this trigger can be interrupted by any trigger including itself.

H - Holdable

Execution of this trigger continues for as long as the trigger is present, repeating itself if necessary. The execution stops immediately when the trigger is no longer present.

In the DS and RS modes, there are 8 direct triggers (001 ~ 008, corresponding to T1 ~ T8). In the BS mode, there are 128 direct triggers (001 ~ 128, corresponding to the binary code on T1 ~ T7).

In addition to the direct triggers, there are also indirect triggers. An indirect trigger can be activated only by jumping from another trigger using the Jump command.

In the DS and RS modes, there are 991 indirect triggers (009 ~ 999). In the BS mode, there are 871 indirect triggers (129 ~ 999).

These are the script commands:

Fnnn - play File #nnn

Example: F168 plays file #168.

Wnnnnn - wait nnnnn units of 0.1 second

Maximum value for nnnnn is 65535 (6553.5 seconds).

Example: W00020 waits 2 seconds.

Jnnn - jump to trigger #nnn

Example: J007 jumps to trigger 007.

BF - turn off the Busy output

Use this command to turn the Busy output off.

BN - turn on the Busy output

Use this command to turn the Busy output on.

XNn - turn on relay #n (EM3028B-R & EM3028B-X only)

Example: XN8 turns on relay #8.

XNN - turn on all relays (EM3028B-R & EM3028B-X only)

Example: XNN turns on all relays.

XFn - turn off relay #n (EM3028B-R & EM3028B-X only)

Example: XF6 turns off relay #6.

XFF - turn off all relays (EM3028B-R & EM3028B-X only)

Example: XFF turns off all relays.

Xnnn - send binary to relay (EM3028B-R & EM3028B-X only)

Example: X63 turns on relay #1~6 because 63 = binary 00111111.

END

Always add the word END at the end of the entire script. You may add any comments for your own reference after END.

Important Notes

- All command letters must be in upper case.
- Script lines must be separated by carriage returns (the Enter key).
- A script line is limited to 128 characters, excluding '=' and ';'. If more space is needed, use the Jump command.

Automatic Execution of Script 000

Upon powerup or reset, the system will automatically executes script 000 once if it exists.

Script Examples

```
DS
N001=F007,W00030,BN,F899,BF,J168
I168=F001,W36000,J168
H033=F273
END
```

DS is not really a script command, but it tells the system to enter the Direct Script mode. You can also use RS or BS to enter the Round-Robin Script mode or the Binary Script mode.

When the T1 input is triggered, the system start executing trigger N001. Since this trigger is non-interruptible, it will always executes to the end. Trigger N001 is executed as the following:

- play file #007,
- wait 3 seconds,
- turn on the Busy output,
- play file #899,
- turn off the Busy output,
- jump to trigger 168 (I168).

Trigger I168 is executed as the following:

- play file #001,
- wait 60 minutes,
- jump back to itself.

Since trigger I168 is interruptible, this endless loop can be broken by any future trigger.

Trigger H033 will never be executed because it is an indirect trigger in the DS mode, yet it is not jumped to by any other trigger.

Background Music Example

The automatic execution feature can be used to play background music while no trigger is being executed. For example,

```
DS
I000=F123,J000
N001=F001,J000
N002=F002,J000
N003=F003,J000
END
```

Here file #123 is looped from power-up but can be interrupted by trigger inputs T1, T2, and T3. After interruption, it always jumps to trigger 000 so file #123 starts to loop again, although from the beginning instead of where it left off.

System Configuration File

By default, the system works in the following mode:

Parallel Control
Direct Trigger
Non-Interruptible Playback
Close Contact

To operate the system in other modes, you need to create a plain text file called "MODE.TXT" with a line of three letters:

First Letter: Trigger Mode

D = Direct
B = Binary
S = Sequential
R = Round Robin

Second Letter: Playback Mode

N = Non-interruptible
I = Interruptible
H = Holdable
S = Script

Third Letter: Trigger Polarity

C (or no letter) = Close Contact
O = Open Contact
M = Make Contact
B = Break Contact

For serial control (RS-232/RS-485), instead of the three letters put a two-digit address ranging from "01" to "32".

For Script modes, enter the script starting from the second line. Be sure to add the word END at the end of the Script.

After editing the configuration file, be sure to save it as a "plain text file", "ASCII text file", or simply "text file". The system may not work if the configuration file is not created properly.

File Number Assignment

Sound files on the flash card must be assigned a unique file number for identification purpose. The file number must be a three digit number within the following range:

For Direct Trigger: 001 ~ 008

For Binary Trigger: 001 ~ 128

For Sequential Trigger and Serial Control: 001 ~ 999 (but limited to 512 files max.)

Simply add the file number to the beginning of the original filename, e.g. "001tiger.wav". Note that if you want to store the maximum number of files (512) on a single flash card, you should keep the filenames (including the file number) within 8 characters. Any filename longer than 8 characters will reduce the number of files you can store on the flash card.

Trouble Shooting Guide

1. Plays no sound at all.

- File numbers are not assigned properly.
- The system is in the wrong mode due to missing or incorrect configuration file.
- If the flash card is inserted when the power is on, the system may not work. To fix this problem, turn the power off for a few seconds to reset the system.
- Some CF cards, especially if they have been used in digital cameras, need to be reformatted with the FAT16 file system.
- The output volume may have been set too low. Try turning it up.

2. Plays a wrong File.

- File numbers are not assigned properly.
- The system is in the wrong mode due to missing or incorrect configuration file.

3. Plays trashy sounds.

It's probably due to unsupported file formats such as 8-bit resolution or ADPCM coding. Re-digitize or convert the file into a supported format.

4. Playback speed is too slow/fast.

Unsupported sampling rates may result in wrong playback speed. Re-digitize or convert the file into a supported sampling rate.

5. Plays a popping/bursting noise once in a while.

The speed of the flash card is too slow. Use a faster flash card or convert the file to a lower sampling rate.

Serial Interface

When the serial interface is enabled, all parallel inputs T1 ~ T8 are disabled and all Parallel Modes are no longer applicable. But there is a serial mode command to read the parallel inputs.

The serial interface consists of a female DB9 serial port supporting both RS-232 (default setting) and RS-485. To support RS-485, you must move the internal jumper J2 to the "485" setting.

On the DB9 connector, three pins are used for RS-232: pin 2 for RX, pin 3 for TX, pin 5 for ground. Two pins on the same connector are used for RS-485: pin 1 for negative, pin 9 for positive.

The hardware protocol is fixed at 9600 baud, eight data bits, no parity and one stop bit (9600, 8N1). Other protocols may be supported by special request.

For RS-232 applications, the system (a DCE device) is connected to a DTE device (such as a PC) with a regular serial cable (not a null modem cable). For RS-485 applications, up to 32 systems, each assigned with a unique address, can be daisy chained on the same bus with a 2-wire cable.

To enable the Serial Mode on the system, the configuration file must contain a two-digit address ranging from "01" to "32". This address assignment seems unnecessary for RS-232 applications because RS-232 is a one-to-one interface. But it is necessary for us because we want to provide a uniform protocol for both RS-232 and RS-485.

The benefit of a uniform protocol is that it allows us to control multiple systems from a DTE which has only a RS-232 port. In this case a RS-232 to RS-485 hardware converter is needed to convert the DTE's RS-232 port into a RS-485 port.

The communication protocol uses software handshake on a per-byte basis. That is, for every byte it receives, the system sends a confirmation byte to the DTE. The DTE must not send the next byte until it receives the confirmation (one exception explained below).

A communication session always starts with a selection process. The DTE should first sends out an ASCII "A" which, as the only exception, will not be confirmed by any system on the bus. The DTE should then sends a binary coded address byte ranging from 1 to 32. If there is a system with the matching address, it will respond by sending back an ASCII "a" within 100 ms. Otherwise the selection process has failed and should be restarted.

Once the selection process is finished successfully, the DTE can issue one of the serial commands, one byte at a time. For each byte sent, the DTE should expect to receive a proper confirmation within 100 ms. If the confirmation is missing or invalid, the whole session must be aborted.

Testing the Serial Port

The serial port can be easily tested by using a Windows utility program called "HyperTerminal". HyperTerminal allows you to send and receive data through the PC's serial port. All you need to do is type the letters on the keyboard to send them, and watch the screen for received data.

The first step is to create a text file called "MODE.TXT" on the flash card. Put only two letters in the file: 01. This will put the player into the Serial mode of operation with a device ID #1. You may use Windows Notepad or Wordpad to do it, just be sure to save the file as "text only" and Windows will add ".TXT" automatically.

In addition to the MODE.TXT file, you also need to put a test sound file on the flash card. For testing purpose, this sound file should be called "001.WAV".

The next step is to connect the player to the PC using a "straight through" serial cable, not a "null modem" one. A straight through cable has pin 2 connected to pin 2, pin 3 connected to pin 3, and pin 5 connected to pin 5. Those are the only three pins we use. Other pins, whether connected or not, do not matter.

Power up the player and run HyperTerminal on the PC. This program is usually found under

Start>Programs>Accessories>Communications>HyperTerminal

Make a new connection on the COM port to which the player is connected, with the following parameters:

```
Bits per second = 9600
Data bits = 8
Parity = None
Stop bits = 1
Flow control = None
```

Now we are ready to send the following commands to the player.

<i>We Type</i>	<i>Player Responds</i>	
A	-	
ALT/01	a	(see notes below)
F	f	
0	0	
0	0	
1	1	

At this point the player should start playing the test sound file, and the test is considered successful.

Note that in order to type the device ID in the binary format on the PC's keyboard, you must hold down the ALT key while entering the numbers on the numerical keypad (make sure the NUM light is on). In this test, for example, we need to type device ID 01, so you must hold the ALT key down, press "0", "1" on the numeric keypad ("0" first then "1"), then release the ALK key.

If you made a mistake and typed a wrong letter during the test, the player will either respond with a letter 'e' (for error) or not respond at all, depending on the situation. In this case, you need to re-type the command from the very beginning.

Serial Commands

The communication protocol uses software handshake on a per-byte basis. That is, for every byte it receives, the system sends an confirmation byte to the DTE. The DTE must not send the next byte until it receives the confirmation byte.

Play File

DTE Sends: F### (### is the three-digit file number)

System Confirms: f### (### is the same file number as above)

If the file exists, it will be played once. If the file does not exist, the system will simply ignore the command.

If the system is playing/paused when the 'F' letter is received, it will return the error code 'e' instead of 'f'. At this point the command should be aborted. You should use the *Stop Playback* command to stop the current playback first before starting a new one.

Loop File

DTE Sends: L### (### is the three-digit file number)

System Confirms: l### (### is the same file number as above)

If the file exists, it will be played repeatedly. If the file does not exist, the system will simply ignore the command.

If the system is playing/paused when the 'L' letter is received, it will return the error code 'e' instead of 'l' (lower case L). At this point the command should be aborted. You should use the *Stop Playback* command to stop the current playback first before starting a new one.

Stop Playback

DTE Sends: S

System Confirms: s

If the system is not playing/paused, it will simply ignore the command. Otherwise it will terminate the current playback.

Pause Playback

DTE Sends: P

System Confirms: p

If the system is not playing, it will simply ignore the command.

When the system is being paused, its Busy output (terminal BY) is still active.

Resume Playback

DTE Sends: R

System Confirms: r

If the system is not paused, it will simply ignore the command.

Busy?

DTE Sends: B

System Confirms: b (if playing/paused) or s (otherwise)

Query Parallel Input Status

DTE Sends: Q

System Confirms: one status byte read from the parallel input

This command is usually used when the parallel inputs are connected to switches, sensors and/or other similar devices, and you want to know the on/off status of these devices. Note that, although the status of these devices can be read, they cannot be used to trigger the audio because the Parallel Interface is disabled when the Serial Port is enabled.

The status byte has T1 as the LSB and T8 as the MSB. Note that an input is at logic "1" when connected to +5V or left unconnected,

and it is at logic "0" when connected to 0V (the ground).

This command can be issued at any time.

Script Command

DTE Sends: C###C (### is one line of script commands)

System Confirms: c###c (### is an echo of the script line)

For example, if you send "CF001W00020F002C" then the system would confirm with "cf001w00020f002c" as the characters are being received. Upon receipt of the second 'C', the system would execute this script line as described below: play file 001 (F001), wait 2 seconds (W00020), play file 002 (F002).

Refer to the 'Script Playback Mode' section for details on how to write the script.

Relay On (EM3038A-R & EM3038A-X only)

DTE Sends: XN# (# is the relay number)

System Confirms: xn# (# is the same relay number as above)

For example, "XN5" turns on relay #5. To turn on all relays, use "XNN". This command can be issued at any time.

Relay Off (EM3038A-R & EM3038A-X only)

DTE Sends: XF# (# is the relay number)

System Confirms: xf# (# is the same relay number as above)

For example, "XF8" turns off relay #8. To turn off all relays, use "XFF". This command can be issued at any time.

Send Binary to Relay (EM3038A-R & EM3038A-X only)

DTE Sends: Xnnn (nnn is a 3-digit number with leading zeros)

System Confirms: xnnn (nnn is the same number as above)

For example, "X005" turns on relay #1 and #3 and turns off all others because the binary code for 005 is 00000101.

Serial Command Error Code

The system will confirm with an "e" if an invalid command is received, or if a valid command is received at the wrong time. At this point, the command should be aborted.

Trouble Shooting Guide

1. The DTE device receives strange characters.

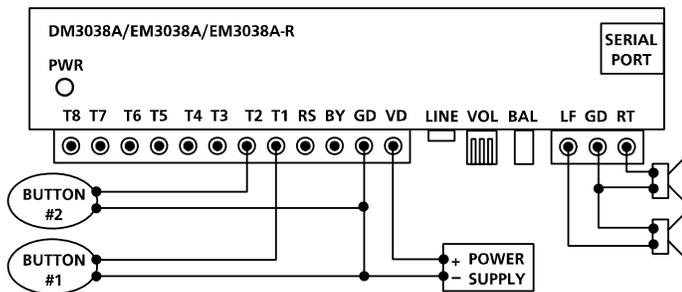
Make sure the DTE device's serial port setting is 9600 baud, 8 data bits, no parity, 1 stop bit.

2. RS-232 does not support multiple unit addressing. Does the DTE device still need to send the address byte after sending the initial letter "A"?

Yes. It is because the protocol is designed to be uniform for both RS-232 and RS-485.

Application Example

Using Normally Open Push Buttons



1. Play without interruption.

Intended Operation

- Press button #1 to play file #001 once.
- Press button #2 to play file #002 once.
- When playing, pressing any button has no effect.

Required Text in MODE.TXT

None, this is the default mode (DN).

Notes

- The button can be released or held down when playing.
- If the button is held down at the end of the sound, the sound will be played again.
- If both buttons are pressed or held down at the same time, button #1 prevails.

2. Allow interruption during playback.

Intended Operation

- Press button #1 to play file #001 once.
- Press button #2 to play file #002 once.
- Playing can be interrupted by pressing any other button.

Required Text in MODE.TXT

DI

Notes

- When playing, pressing the same button again has no effect.

3. Play the sound only once even if the button is held down.

Intended Operation

- Press button #1 to play file #001.
- Press button #2 to play file #002.
- Don't repeat the sound even if the button is held down.

Required Text in MODE.TXT

DS

N001=F001,J888

N002=F002,J888

H888=J888

END

Notes

- The system will stay in the "H888=J888" endless loop for as long as the button is held down.

4. Play only when the button is held down.

Intended Operation

- Press & hold button #1 to play file #001.
- Press & hold button #2 to play file #002.

Required Text in MODE.TXT

DH

Notes

- Playing stops as soon as the button is released. If the same button is pressed again later, playing re-starts from the beginning of the file instead of where it left off.
- If both buttons are held down, button #1 prevails. To play both sounds alternately when both buttons are held down, use the RH mode.

5. Play different sounds each time when the same button is pressed.

Intended Operation

- Press button #1 to play file #001 the first time, file #002 the second time, and etc.
- Button #2 sequences through file #201, #202...
- When playing, pressing any button has no effect.

Required Text in MODE.TXT

SN

Notes

- Up to 99 files can be assigned to each button and file numbers but be consecutive.
- To allow playback interruption, use the SI mode.
- To play only when the button is held down, use the SH mode. In the SH mode, the same sound will repeat for as long as the button is held down. To advance to the next sound, the button must be released first.

6. EM3028B-R: Turn on a different relay for each sound.

Intended Operation

- Turn on relay #1 when playing file #001.
- Turn on relay #2 when playing file #002.

Required Text in MODE.TXT

Only the mode letters are required, for example: DH.

7. EM3028B-R: Turn on multiple relays for each sound.

Intended Operation

- Press button #1 to play file #007 and turn on relays #1 & 2.
- Press button #2 to play file #008 and turn on relays #2 & 3.

Required Text in MODE.TXT

DS

N001=XN1,XN2,F007,XFF

N002=XN2,XN3,F008,XFF

END

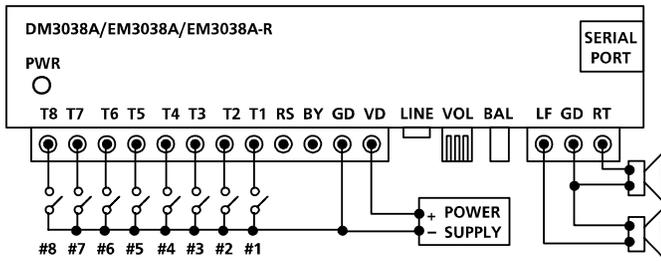
Notes

- XFF is used to turn off all relays.

8. EM3028B-X: Control relays during audio playback.

The relays will be controlled automatically when playing QWA files. There is no required text in MODE.TXT.

Application Example
Automatic Playback on Power-up



1. Repeat continuously on power-up.

Intended Operation

- Play file #001 on power-up if switch #1 is turned on.
- Play file #002 on power-up if switch #2 is turned on.
-
- Play file #008 on power-up if switch #8 is turned on.
- If more than one switch is turned on, play all corresponding files sequentially
- Repeat the file(s) until power is turned off.

Required Text in MODE.TXT

RN

Notes

- A direct short can be used if there is only one sound file.
- If priority is required, use DN instead of RN. In this case switch #1 has the highest priority, switch #2 has the second highest priority, and switch #8 has the lowest priority. If more than one switch is turned on, only the file for the switch of the highest priority will be played.

2. Repeat at intervals on power-up.

Intended Operation

- Same as the example above but play files at intervals.

Required Text in MODE.TXT

RS

N001=F001,W06000

N002=F002,W06000

.....

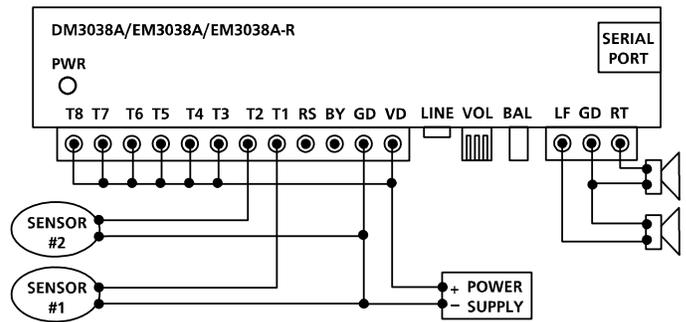
N008=F008,W06000

END

Notes

- W06000 is a delay loop of 6000 x 0.1 second = 10 minutes, so the file(s) will be played at an interval of 10 minutes, one file at a time.

Application Example
Using Normally Closed Switches & Sensors



Normally closed switches and sensors are often used in security systems.

1. Play when the switch/sensor opens, with no priorities.

Intended Operation

- Play file #001 when sensor #1 opens.
- Play file #002 when sensor #2 opens.
- Repeat the sound for as long as the sensor is open.
- If multiple sensors are open at the same time, play all corresponding files sequentially.

Required Text in MODE.TXT

RN+

Notes

- Unused inputs must be connected to the ground, as shown in the wiring diagram.

2. Play when the switch/sensor opens, with priorities.

Intended Operation

- Same as the example above, but if multiple sensors are open at the same time, the sensor of the highest priority prevails.

Required Text in MODE.TXT

DN+

Notes

- The sensor connected to T1 has the highest priority, and the sensor connected to T8 has the lowest priority.

3. Play when the switch/sensor opens, but don't repeat.

Intended Operation

- Play file #001 when sensor #1 opens.
- Play file #002 when sensor #2 opens.
- Do not repeat the sound.

Required Text in MODE.TXT

DS+

N001=F001,J999

N002=F002,J999

H999=J999

END

Notes

- The system will stay in the "H999=J999" endless loop for as long as the sensor is open. The system will not respond to other sensors until this one is closed.