



## Firmware Version

1.7

## Technical Specifications

### Operation Mode

playback only

### Sound File Format

Windows WAV (uncompressed PCM)  
 - 6/8/11/16/22/32/44 KHz sample rate  
 Use 32/44 KHz for lower background noise.  
 - 8 bit resolution  
 - monaural

### Max. Number of Sound Files

Direct Mode: 8  
 Binary Mode: 128  
 Sequential Mode: 8 x 99  
 Serial Mode: 999

### Memory Card Type

SD/SDHC or MicroSD/MicroSDHC w/adaptor

### Max. Memory Capacity

2 GB for SD/MicroSD (FAT/FAT16)  
 32 GB for SDHC/MicroSDHC (FAT32)

### Max. Recording Time

2GB = 33 hours (16 KHz sample rate)  
 32GB = 528 hours (16 KHz sample rate)

### Supply Voltage

12 ~ 30 VDC

### Typical Standby Current

60 mA

### Audio Output

Amplifier Type: class D, single ended  
 Min. Load: 4 Ohm  
 Max. Output Power: 15W  
 (30V supply, 8 Ohm load, 10% THD+N)

### Serial Interface

RS-232

### Trigger Inputs

8 inputs, contact closure or 3.3V logic

### Physical Dimensions

4.625" x 3.25" x 1.25"

## Inputs, Outputs & Controls

### Power Input Terminals: V+, GD

Connect the power supply's positive end to terminal V+, and the negative end to terminal GD. Use a well regulated power supply for best sound quality.

### Speaker Output Terminals: S1, S2

Polarity is not important but terminal S2 is tied to the ground internally.

### Trigger Input Terminals: T1 - T8

These inputs are internally pulled up to 3.3Vdc and seen as logic "1" if actively driven by a DC signal within the range of 3.3V to 5V (for units purchased before October 2015) or 3.3V to 40V (for units purchased in/after October 2015), or simply left undriven. To be seen as logic "0" these inputs must be actively driven to 0V (ground).

### Power Light (PWR)

Power light is steady green.

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

This output can be used to feed an external amplifier if the internal power amplifier is not adequate for the application. The output level is not affected by the volume pot.

### Volume Pot (VOL)

Turn the volume pot clockwise to increase the output. It affects the speaker output only, not the line output.

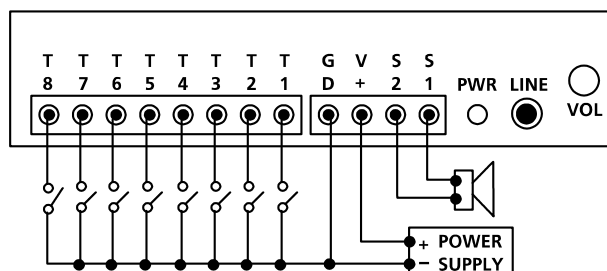
### Serial Port: DB9 female connector

Located on the back of the unit, the RS-232 serial port is used to send control commands, not audio data, to the unit. It is disabled by default. See *System Configuration* on how to enable it.

## Typical wiring diagram for push button activation

The following wiring diagram illustrates the most common application. Up to eight push buttons are connected to the trigger terminals, with each button activating a different message/sound file.

In addition to this default mode of operation, the system can also be configured to other modes of operation by adding a simple text file on the memory card. See *System Configuration* for more information.



## **Parallel Trigger Modes**

Parallel Trigger Mode defines how the playback is to be triggered via the trigger inputs.

### **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 applied 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".

Thanks to internal pull-up, all un-connected inputs are at logic "1" by default.

The second step is to pulse T8 to the ground for at least 50 ms while maintaining the signal on T1 ~ T7. 50 ms after T8 is released from the ground, signals on T1 ~ T7 don't matter anymore and may be released.

### **Sequential Trigger**

Use the Sequential Trigger to sequentially trigger up to 99 different files per input as described below. However, the total number of files cannot exceed 511 unless the memory card is formatted with the FAT32 file system.

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 can interrupt the playback except that an input cannot interrupt itself if the Trigger Polarity is either Closed or Open.

### **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**

### **Closed Contact**

Input is continuously triggered when it's at 0V (ground).

### **Open Contact**

Input is continuously triggered when it's left open or at 3.3V/5V.

### **Make Contact**

Input is triggered one time as it goes from 3.3V/5V to 0V.

### **Break Contact**

Input is triggered one time as it goes from 0V to 3.3V/5V.

## Script Playback Mode

Instead of playing a single sound file, the 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 specifically turned on/off with the BN and the BF commands.

Written in the configuration file using plain 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 other than itself.

### **H - Holdable**

Execution of this trigger continues for as long as the trigger is applied, repeating itself if necessary. The execution stops immediately when the trigger is removed.

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 = wait 2 seconds.

Note: W00000 = wait forever.

### **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.

### **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**

To play background music (BGM) automatically when the system is idle, just number the BGM file A01 (e.g. A01.MP3). To play a list of BGM files, just number them A01, A02 ... (up to A99). After interruption, BGM will resume from where it left off.

## **System Configuration File**

By default, the system works in the following mode without a configuration file:

*Trigger Input: Parallel*  
*Trigger Mode: Direct*  
*Playback Mode: Non-Interruptible*  
*Trigger Polarity: Closed Contact*

To operate the system in any 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) = Closed Contact  
O = Open Contact  
M = Make Contact  
B = Break Contact

To enable the serial port (RS-232/RS-485), instead of the three letters described above, put only a two-digit number 00. This enables the serial port and configures the unit into the serial mode.

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

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 on the flash card, you should keep the filenames (including the file number) within 8 characters and use numbers and capital letters only.

## **Trouble Shooting Guide**

### **1. Plays no sound at all.**

- a. File numbers are not assigned properly.
- b. The system is in the wrong mode due to missing or incorrect configuration file.
- c. 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.
- d. Some CF cards, especially if they have been used in digital cameras, need to be reformatted with the FAT16 file system.
- e. The output volume may have been set too low. Try turning it up.

### **2. Plays a wrong File.**

- a. File numbers are not assigned properly.
- b. 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 16-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.

### **6. There is a high level of background noise.**

Try converting the sound to a higher sampling rate such as 32 or 44 KHz. However, a faster flash card may be needed to properly play the sound at a higher sampling rate.

### **7. 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.

## Serial Interface

When the serial interface is enabled, all parallel inputs T1 ~ T8 are disabled and all Parallel Modes are no longer available.

The serial interface consists of a female DB9 serial port supporting the RS-232 communication protocol.

On the DB9 connector, only three pins are used: pin 2 for RX, pin 3 for TX, pin 5 for ground.

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.

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).

To enable the Serial Mode on the system, the configuration file must contain the following two digits: 00. These two digits are the only text required for the configuration file.

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, otherwise commands may not be received properly.

A communication session always starts with an initial process. The DTE always initiates the communication by sending an ASCII letter 'A'. The device should respond by sending back an ASCII letter "a" within 100 ms. Otherwise the initial process has failed and should be restarted.

After the initial process the DTE can issue the command one byte at a time. For each byte sent, the DTE should wait to receive a proper confirmation within 100 ms. If the confirmation is missing or invalid, the whole session must be aborted.

The initial process must precede each and every command.

## 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: 00. This will put the system into the serial mode. You may use Windows Notepad or Wordpad to create the file, just be sure to save the file as "text only".

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 system 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. On Windows XP this program is found under

*Start>All Programs>Accessories>Communications*

Create 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>	
<i>A</i>	<i>a</i>	<i>(see notes below)</i>
<i>F</i>	<i>f</i>	
<i>0</i>	<i>0</i>	
<i>0</i>	<i>0</i>	
<i>1</i>	<i>1</i>	

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

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 (letter A).

## **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).

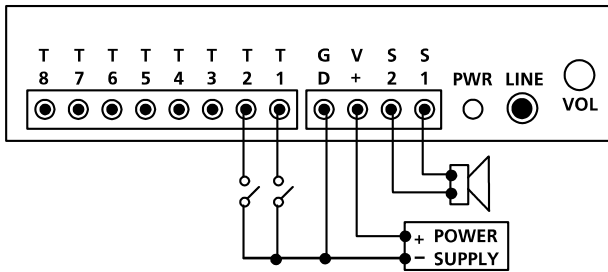
Refer to the 'Script Playback Mode' section for details on how to write the script.

### **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.

**Application Examples**

**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**

DNM

- if playback is to be non-interruptible

DIM

- if playback is to be interruptible

**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 a list of sounds sequentially, one per trigger, 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**

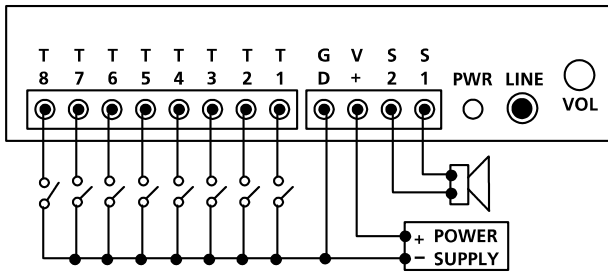
SN

**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.

**Application Examples**

**Automatic Playback on Power-up**



**1. Repeat continuously on power-up, with no priorities.**

**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

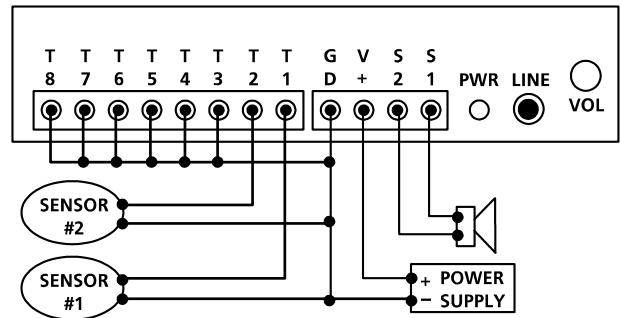
END

**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 Examples**

**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**

RNO

**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**

DNO

**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**

DNB

- if playback is to be non-interruptible

DIB

- if playback is to be interruptible