

Subject: AudioSaver Hardware Setup

Parwan Electronics Corporation
480 Route 9
Englishtown, NJ 07726, USA

Date: 7/30/96

John Albright
Tele: (908)536-7500
Fax: (908)536-1729

Technical Memorandum

1 Introduction

Bingo is an application (19), that AudioSaver will run. It simulates a bingo game over the telephone, and calls different balls to the player. The player then marks these balls on his or her game sheets, and if they get five in a row, they win.

2 Perspective

2.1 Bingo from the Callers Perspective

The caller dials the phone number connected to the mailbox set to run the bingo application. They are then presented with a series of balls which they will mark off on their sheets. If they get a row of five spaces filled in, they are a winner, and can then receive prize information.

2.2 Bingo from the Administrators Perspective

The administrator has the option to run two versions of bingo. One is interactive, where the caller must call out the word "bingo" if he or she completes a row of five spaces. The other, non-interactive version, will automatically detect if the user has bingo, and move them to the prize awarding automatically. There are advantages to both setups. The main difference being that the non-interactive version runs quicker, and takes us less time.

Parwan Electronics Corporation Restricted Information
E-mail: pec@voicesaver.com document: bingo.961

3 Grunt Setup

The following is a grunt table for the non-interactive version of Bingo:

Step	Opening	Grunt	NStep	P1	Ngrunt	NStep	P2
1	INTRO	NULL	2	7	NULL	2	7
2	BALLS	B??00	3	0	B??00	3	0
3	NULL	B??01	2	4	B??01	2	5
4	BINGO		0	0		0	0
5	NOWIN		0	0		0	0
6			0	0		0	0
7			0	0		0	0

Note: P in step 1 set the number of balls for the program to use.

Here is a text outline of what the grunt table will do when run:

Step 1: Play audio file INTRO, and proceed to step 2. P will set number of balls (currently 7).

Step 2: Play file BALLS (An audio file saying "Ball Number..."). Function B??00 is then called for the first time and will play an audio file called BALL01 (saying "... one."), and proceed to step 3.

Step 3: This step will call function B??01. This function, being called for the first time, will randomly pick ball number one. It will then play the corresponding voice file (for example, 01210 for ball 21 {O, 1}). After the voice file is played to the caller, the function will check to see if the user has bingo or if the last ball has been called. Since this is the first ball, the answer to both questions is no, and the application will follow NStep back to step 2.

Step 2: Step 2 will now be repeated and incremented to ball number 2. Step 2 will call "Ball number two.", and proceed to step 3 again.

Step 3: Step three will be repeated for ball 2. Again, picking a random ball and letting the caller know which one has been chosen. It will then recheck for bingo or last ball, and loop back to step 2.

Balls 3 - 7: Steps two and three will be repeated for balls three through seven. If at anytime the application detects that the user has bingo, it will proceed to the step indicated by the Grunt P column in the step containing the function B??01. If ball seven is called and the user still has not gotten bingo, the application will proceed to the step indicated by the NoGrunt P column in the step containing the function B??01.

Step 4: This step will be reached when the application detects that the caller has gotten bingo. It will play the file BINGO (A file containing "Congratulations, you have just won bingo."), and terminate the application because next step is zero.

Step 5: This step will be reached when the application calls all seven balls (or however many are specified by P in step one), and the caller has still not gotten bingo. It will play a file NOWIN (containing voice saying "Sorry, you did not win this time. Please call again."), and will terminate the application because NStep is zero.

There are five functions specific to the bingo application. Below is a list of the functions specific to application 19, what they do, and the parameters that can be specified for each.

B??00 - This function will play a file to announce the ball number. This file must be named BALLbb where bb is a two digit number. This function will keep track of how many times it has been run and play the next file each time it is run. This function has no parameters.

B??01 - This function will use Pool #1 to get a random ball from 1 to 25, excluding 13 (the free space), and play the corresponding voice file, which announces the ball number. The corresponding voice file must be named 01nn0 where nn is the ball number from 1 to 25. Parameter 1, under the Grunt section, will be set to the step to skip to should the user get bingo. Parameter 2, under the NoGrunt section, will be set to the step to skip to when all balls have been called, and the caller has still not gotten bingo.

B??02 - Same as B??01, except it will use Pool #2 for board #2. The file names must be 02nn0, using the same format as function B??01. The parameters are also the same as for function B??01.

B??03 through B??09 - These functions are the same as B??01 and B??02, except they are used for Pool #'s 3-9, and Board #'s 3-9.

B??0R - When this function is called, it will reset all the counts kept by the functions, in order to begin a new game of bingo. This function has no parameters.

The following is a grunt table for the interactive version of bingo:

Step	Opening	Grunt	NStep	P1	Ngrunt	NStep	P2
1	INTRO	NULL	2	0	NULL	2	0
2	BALLS	NULL	3	0	NULL	3	0
3	BALL1	B??01	4	4	B??01	4	4
4	BALLS	NULL	5	0	NULL	5	0
5	BALL2	B??01	6	6	B??01	6	6
6	BALLS	NULL	7	0	NULL	7	0
7	BALL3	B??01	8	8	B??01	8	8
8	BALLS	NULL	9	0	NULL	9	0
9	BALL4	B??01	10	10	B??01	10	10
10	NULL	BINGO	25	0	NULL	11	0
11	BALLS	NULL	12	0	NULL	12	0
12	BALL5	B??01	13	13	B??01	13	13
13	NULL	BINGO	25	0	NULL	14	0
14	BALLS	NULL	15	0	NULL	15	0
15	BALL6	B??01	16	16	B??01	16	16
16	NULL	BINGO	25	0	NULL	17	0
17	BALLS	NULL	18	0	NULL	18	0
18	BALL7	B??01	19	19	B??01	19	19
19	NULL	BINGO	25	0	NULL	20	0
20	NOWIN	NULL	21	0	NULL	21	0
21	AGAIN	NULL	2	0	BYE	0	0
22			0	0		0	0

The functions are the same as in the non-interactive version. The number of balls is not set by a parameter, and large changes will need to be made to change from seven balls. When a caller announces that they have bingo they will be moved to step 25 where prize awarding can be done, and if they draw all seven balls and do not win, they will be asked at step 21 if they wish to play again.

4 Recordings

Recordings should be made for the following files:

BALLS - voice file containing, "Ball number..."

BALLbb - where bb is 01 through the number of balls to be used. This file is used to announce the ball number.

01nn0 - where nn is the space number from 01 to 25. These are voice files containing the coordinates for the space, and the number contained.

For example, "B, three....twenty-five".

INTRO - a voice file containing an introduction to the game

BINGO - a voice file containing a message to be played when the caller gets bingo.

NOWIN - a voice file containing a message to be played when the caller has drawn all balls, and has not won.

BALLb - where b is the ball number from 1 through 7. (Only needed in the interactive version)

5 Running the Game

A Mailbox should now be setup in AudioSaver to run and test this application. The proper directory where your grunt table and voice files are stored should be entered in the GruntDir field.

BOARD # 1

(figure 1.)

	B	I	N	G	O
1	5	9	15	4	11
2	7	13	20	2	17
3	19	24	FREE	23	21
4	12	16	1	10	8
5	3	22	18	6	14

BOARD # 2

	B	I	N	G	O
1	10	3	8	19	18
2	2	22	15	7	12
3	4	24	FREE	14	23
4	1	6	17	5	20
5	9	13	11	16	21

BOARD # 3

	B	I	N	G	O
1	24	9	15	4	20
2	7	13	11	2	17
3	1	5	FREE	23	21
4	12	16	19	10	8
5	3	22	18	6	14

BOARD # 4

	B	I	N	G	O
1	14	13	23	18	3
2	7	9	20	6	21
3	12	22	FREE	15	8
4	19	16	1	10	17
5	11	24	4	2	5