

Features

- Operating voltage: 2.4V~5.0 V
- Programmable tone melody generator
- ADPCM, μ PCM, PCM synthesis
- Wide range of sample rate for voice synthesis
- Minimum sample rate step: 100Hz
- Voice melody mixed output
- Programmable 2 channels of melody mixed output
- Programmable 2 channels of voice mixed output
- 13 kinds of melody beats
- 5 octaves of tone level and 18 tempos
- 16 levels of digital volume control
- Two current type D/A outputs
- Eight programmable I/O pins
- Four programmable input pins
- Provides 3sec to 384sec of voice capacity
- Timer controller
- Voice fill_in function
- Powerful user-defined functions
- Power-on initial setting

Applications

- High-end educational toys
- Alert and warning systems
- Speech synthesizers and sound effect generators

General Description

The HT84XXX family is a series of programmable speech synthesizers and tone generators designed for user-defined voice and melody applications. It provides various sample rates and beats, tone levels, tempos for the speech synthesizer and melody generator. The HT84XXX series has a built-in two high quality, current type D/A outputs with 16 levels of volume control.

The user's commands enable the user to program the powerful custom function such as to build-in a tiny controller. The Magic Voice™ provides various operational functions similar to arithmetic operation, logic operation, branch decision, random counter, and a programmable timer. The Magic Voice™ series are suitable for versatile voice and sound effect applications.

ROM Selection Table

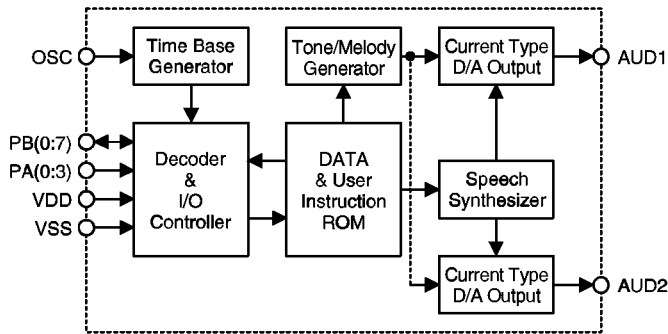
The HT84XXX series provides various voice capacity as shown below:

Part No.	HT84003	HT84006	HT84009	HT84012	HT84018	HT84024	HT84036	HT84048
ROM	8KB	16KB	24KB	32KB	48KB	64KB	96KB	128KB
Voice Length	3 sec	6 sec	9 sec	12 sec	18 sec	24 sec	36 sec	48 sec

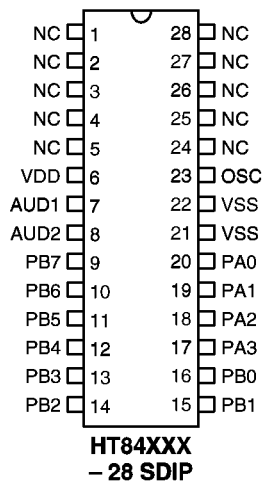
Part No.	HT84072	HT84096	HT84144	HT84192	HT84240	HT84288	HT84336	HT84384
ROM	192KB	256KB	384KB	512KB	640KB	768KB	896KB	1024KB
Voice Length	72 sec	96 sec	144 sec	192 sec	240 sec	288 sec	336 sec	384 sec

Note: The voice capacity is based on a sample rate of 21kb/s

Block Diagram

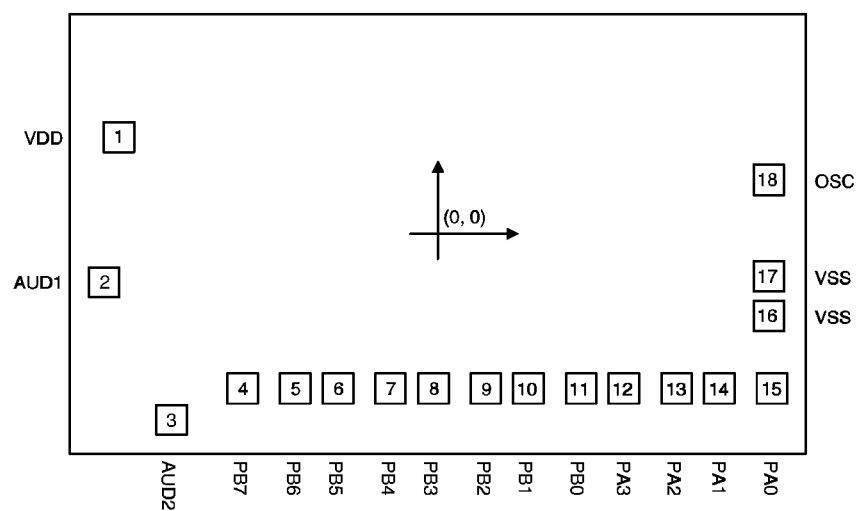


Pin Assignment



Fosc	IC Rosc	EV Rosc
4M	240K	180K
5M	210K	150K
6M	180K	120K
8M	150K	91K

Pad Assignment (HT84003, HT84006)



Chip size: $2735 \times 1650 (\mu\text{m})^2$

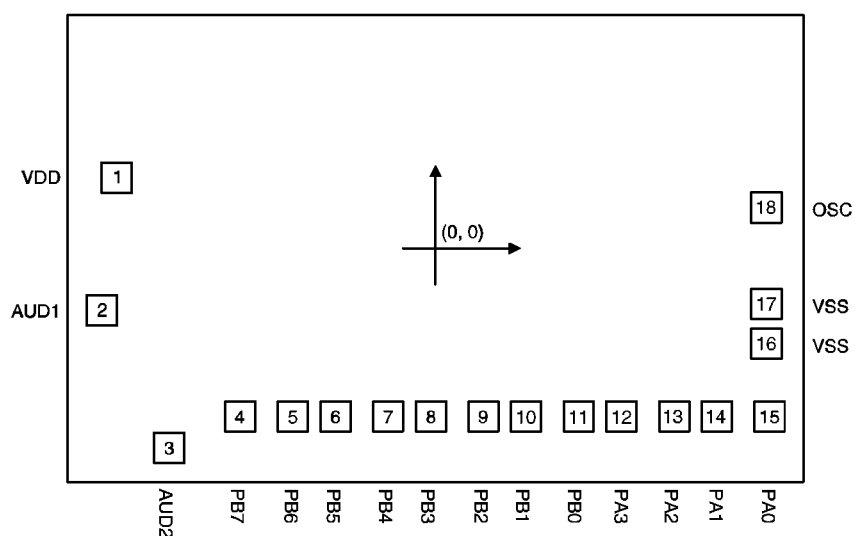
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	342.24	10	320.15	-546.74
2	-1177.49	-171.29	11	505.75	-546.74
3	-939.68	-657.11	12	656.55	-546.74
4	-689.05	-546.74	13	842.15	-546.74
5	-503.45	-546.74	14	992.95	-546.74
6	-352.65	-546.74	15	1178.55	-546.74
7	-167.05	-546.74	16	1167.43	-289.68
8	-16.25	-546.74	17	1167.53	-149.58
9	169.35	-546.74	18	1167.53	191.54

Pad Assignment (HT84009, HT84012)



Chip size: $2735 \times 1745 (\mu\text{m})^2$

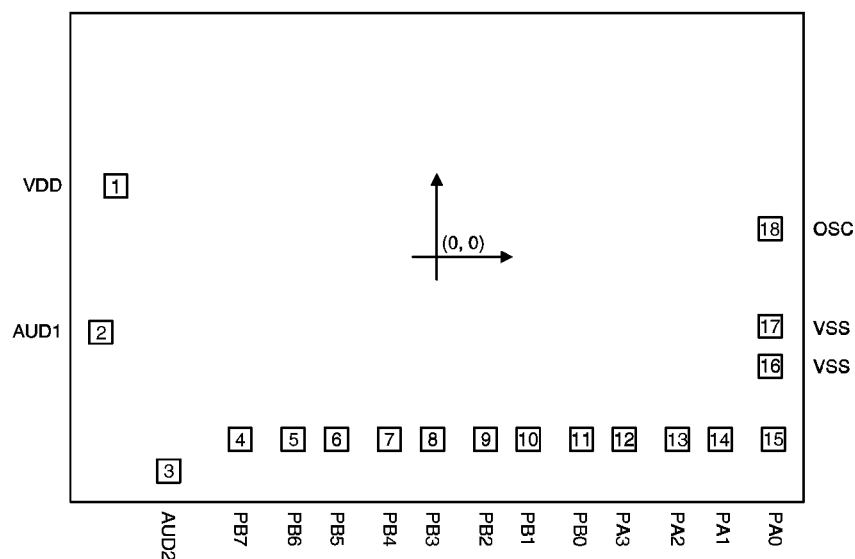
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	249.74	10	320.15	-594.24
2	-1177.49	-218.79	11	505.75	-594.24
3	-939.68	-704.61	12	656.55	-594.24
4	-689.05	-594.24	13	842.15	-594.24
5	-503.45	-594.24	14	992.95	-594.24
6	-352.65	-594.24	15	1178.55	-594.24
7	-167.05	-594.24	16	1167.43	-337.18
8	-16.25	-594.24	17	1167.53	-197.08
9	169.35	-594.24	18	1167.53	144.04

Pad Assignment (HT84018)



Chip size: $2735 \times 1835 (\mu\text{m})^2$

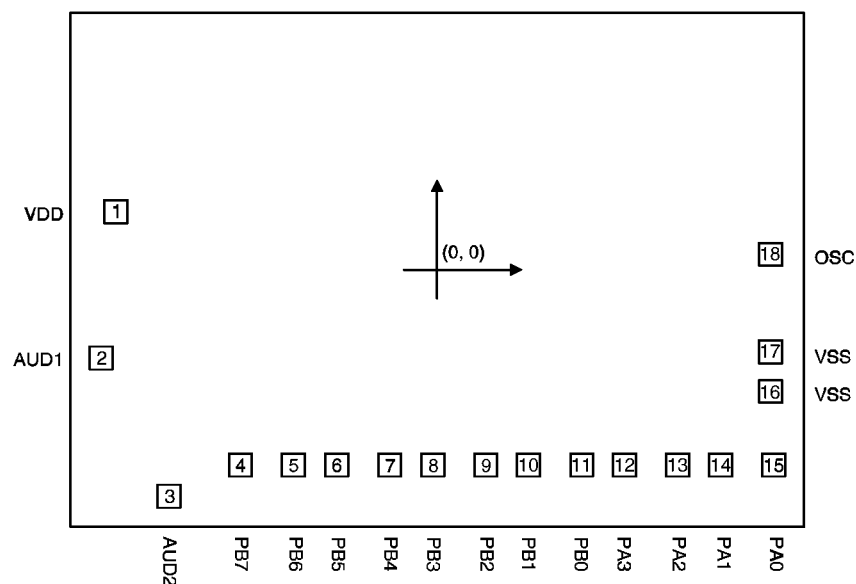
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	249.74	10	320.15	-639.24
2	-1177.49	-263.79	11	505.75	-639.24
3	-939.68	-749.61	12	656.55	-639.24
4	-689.05	-639.24	13	842.15	-639.24
5	-503.45	-639.24	14	992.95	-639.24
6	-352.65	-639.24	15	1178.55	-639.24
7	-167.05	-639.24	16	1167.43	-382.18
8	-16.25	-639.24	17	1167.53	-242.08
9	169.35	-639.24	18	1167.53	99.04

Pad Assignment (HT84024)



Chip size: $2735 \times 1925 (\mu\text{m})^2$

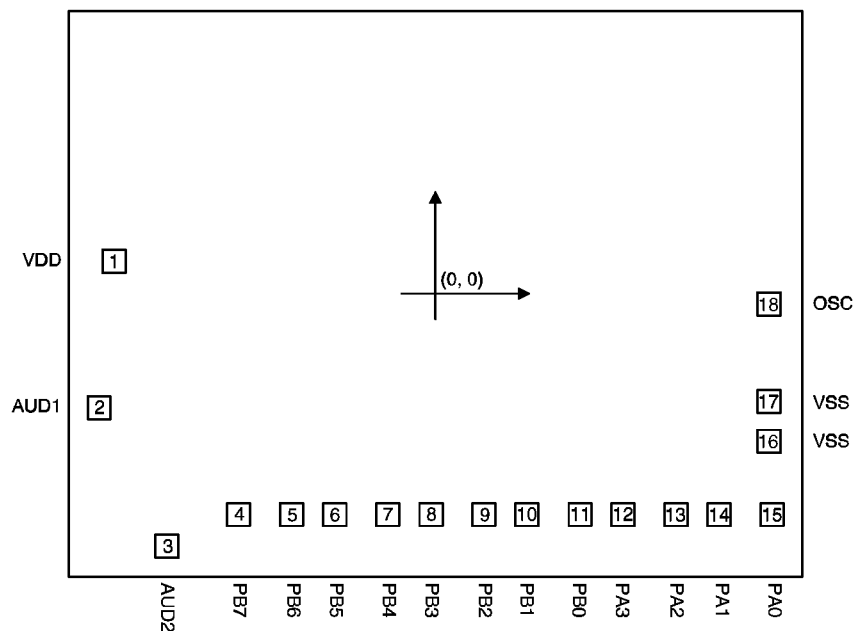
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	204.74	10	320.15	-684.24
2	-1177.49	-308.79	11	505.75	-684.24
3	-939.68	-794.61	12	656.55	-684.24
4	-689.05	-684.24	13	842.15	-684.24
5	-503.45	-684.24	14	992.95	-684.24
6	-352.65	-684.24	15	1178.55	-684.24
7	-167.05	-684.24	16	1167.43	-427.18
8	-16.25	-684.24	17	1167.53	-287.08
9	169.35	-684.24	18	1167.53	54.04

Pad Assignment (HT84036)



Chip size: $2735 \times 2105 (\mu\text{m})^2$

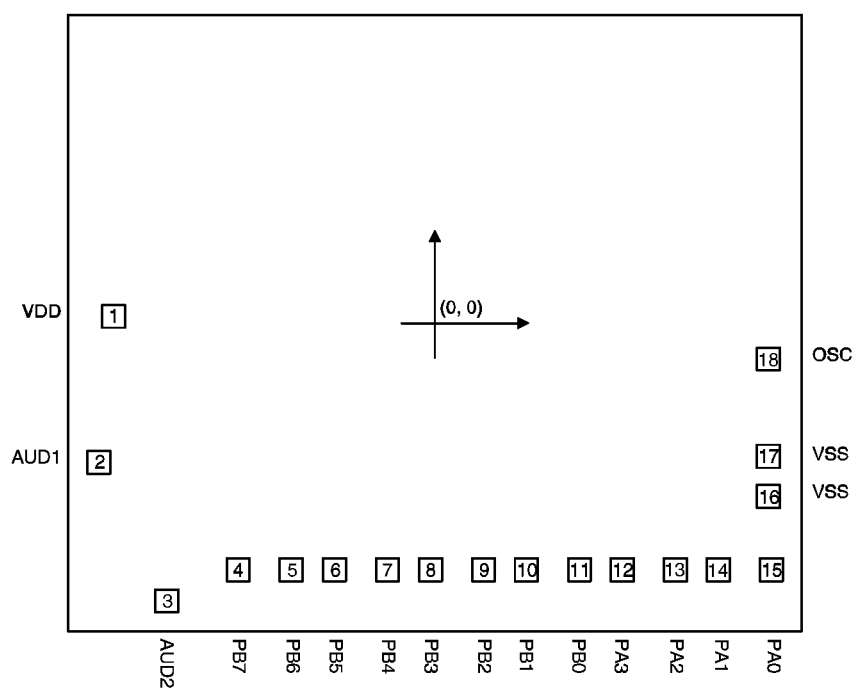
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	114.74	10	320.15	-774.24
2	-1177.49	-398.79	11	505.75	-774.24
3	-939.68	-884.61	12	656.55	-774.24
4	-689.05	-774.24	13	842.15	-774.24
5	-503.45	-774.24	14	992.95	-774.24
6	-352.65	-774.24	15	1178.55	-774.24
7	-167.05	-774.24	16	1167.43	-517.18
8	-16.25	-774.24	17	1167.53	-377.08
9	169.35	-774.24	18	1167.53	-35.96

Pad Assignment (HT84048)



Chip size: $2735 \times 2285 (\mu\text{m})^2$

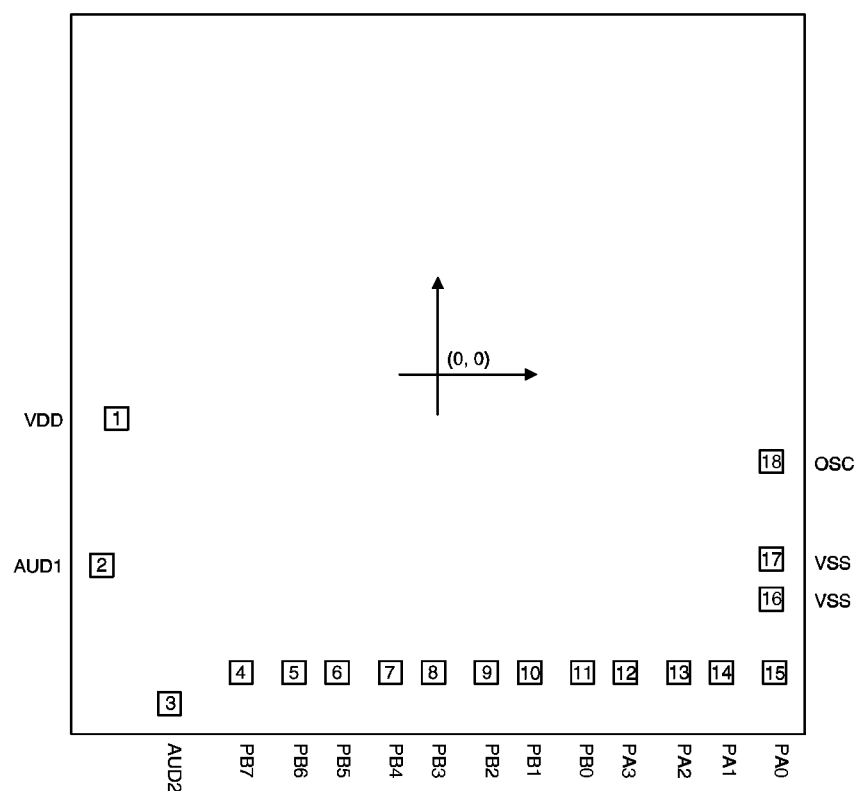
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	24.74	10	320.15	-864.24
2	-1177.49	-488.79	11	505.75	-864.24
3	-939.68	-974.61	12	656.55	-864.24
4	-689.05	-864.24	13	842.15	-864.24
5	-503.45	-864.24	14	992.95	-864.24
6	-352.65	-864.24	15	1178.55	-864.24
7	-167.05	-864.24	16	1167.43	-607.18
8	-16.25	-864.24	17	1167.53	-467.08
9	169.35	-864.24	18	1167.53	-125.96

Pad Assignment (HT84072)



Chip size: $2735 \times 2645 (\mu\text{m})^2$

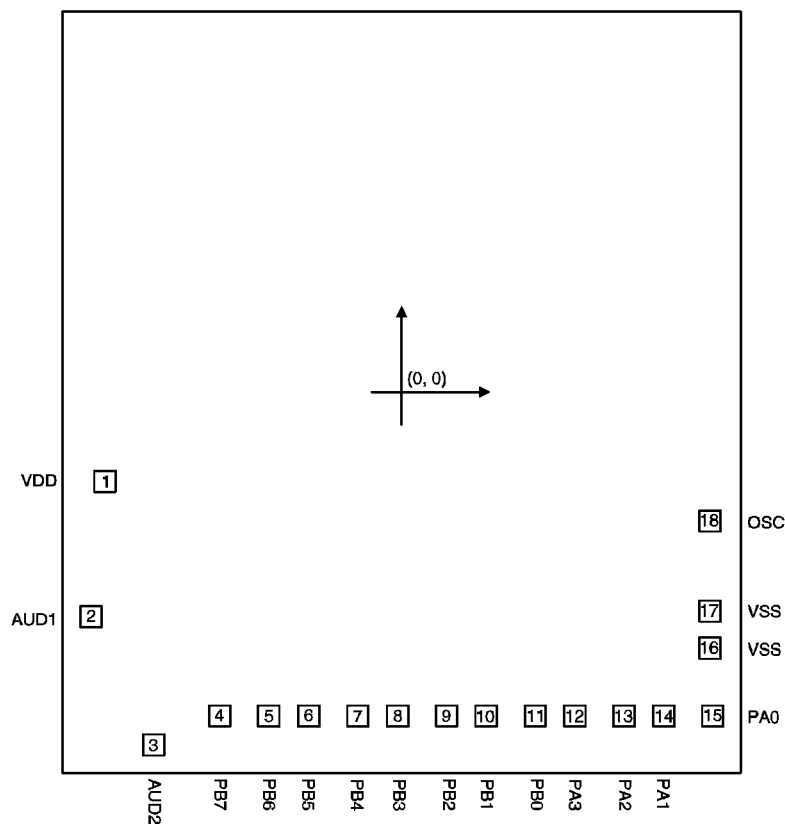
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	-155.26	10	320.15	-1044.24
2	-1177.49	-668.79	11	505.75	-1044.24
3	-939.68	-1154.61	12	656.55	-1044.24
4	-689.05	-1044.24	13	842.15	-1044.24
5	-503.45	-1044.24	14	992.95	-1044.24
6	-352.65	-1044.24	15	1178.55	-1044.24
7	-167.05	-1044.24	16	1167.43	-787.18
8	-16.25	-1044.24	17	1167.53	-647.08
9	169.35	-1044.24	18	1167.53	-305.96

Pad Assignment (HT84096)



Chip size: $2735 \times 3010 (\mu\text{m})^2$

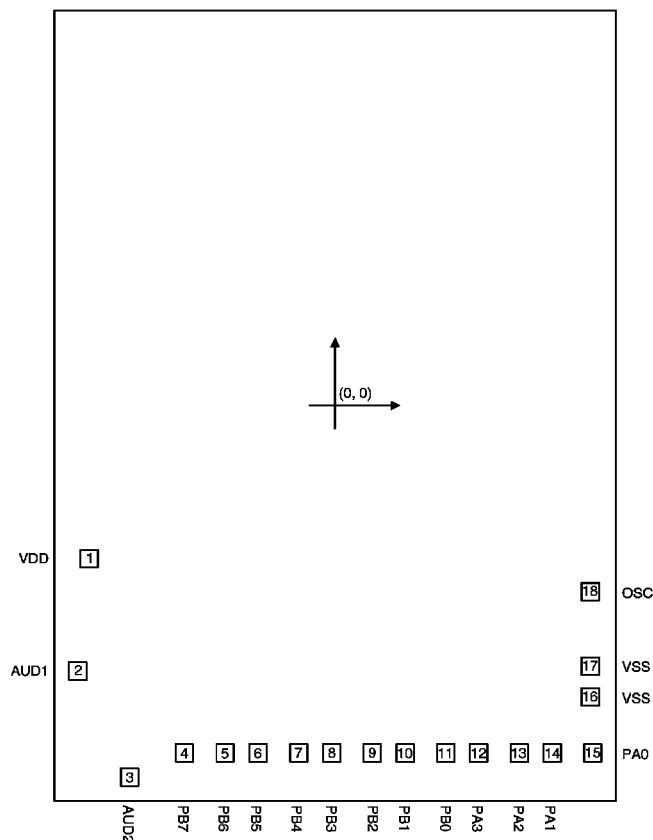
* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	-337.76	10	320.15	-1226.74
2	-1177.49	-851.29	11	505.75	-1226.74
3	-939.68	-1337.11	12	656.55	-1226.74
4	-689.05	-1226.74	13	842.15	-1226.74
5	-503.45	-1226.74	14	992.95	-1226.74
6	-352.65	-1226.74	15	1178.55	-1226.74
7	-167.05	-1226.74	16	1167.43	-969.68
8	-16.25	-1226.74	17	1167.53	-829.58
9	169.35	-1226.74	18	1167.53	-488.46

Pad Assignment (HT84144)



Chip size: 2735 × 3735 (μm)²

* The IC substrate should be connected to VSS in the PCB layout artwork.

Pad Coordinates

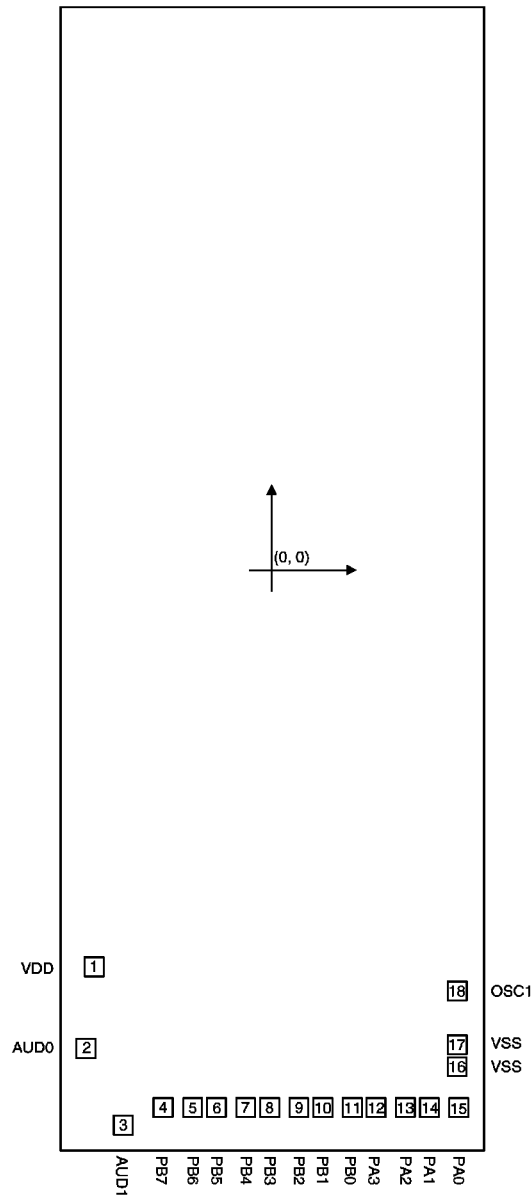
Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	-700.26	10	320.15	-1589.24
2	-1177.49	-1213.79	11	505.75	-1589.24
3	-939.68	-1699.61	12	656.55	-1589.24
4	-689.05	-1589.24	13	842.15	-1589.24
5	-503.45	-1589.24	14	992.95	-1589.24
6	-352.65	-1589.24	15	1178.55	-1589.24
7	-167.05	-1589.24	16	1167.43	-1332.18
8	-16.25	-1589.24	17	1167.53	-1192.08
9	169.35	-1589.24	18	1167.53	-850.96

Pad Assignment (HT84384)

Pad Coordinates

Unit: μm



Pad No.	X	Y
1	-1121.81	-2507.96
2	-1173.99	-3021.49
3	-936.18	-3507.31
4	-685.55	-3396.94
5	-499.95	-3396.94
6	-349.15	-3396.94
7	-163.55	-3396.94
8	-12.75	-3396.94
9	172.85	-3396.94
10	323.65	-3396.94
11	509.25	-3396.94
12	660.05	-3396.94
13	845.65	-3396.94
14	996.45	-3396.94
15	1182.05	-3396.94
16	1170.93	-3139.88
17	1171.03	-2999.78
18	1171.03	-2658.66

Chip size: $2740 \times 7350 (\mu\text{m})^2$

* The IC substrate should be connected to VSS in the PCB layout artwork.

Pin Description

Pin Name	I/O	Internal Connection	Description
PA (0:3)	I	Wake-up Pull-High	Trigger inputs Can also be configured as wake-up inputs
PB (1:7)	I/O	Pull-High or CMOS	Bidirectional I/O pins Can be optioned as trigger inputs or LED outputs
PB0	I/O	Pull-High or CMOS	Bidirectional I/O pins Can be optioned as CDS interface with internal Schmitt trigger input
AUD1	O	PMOS Open Drain	Audio output for driving an external transistor
AUD2	O	PMOS Open Drain	Audio output for driving an external transistor
OSC	I	—	Built-in RC oscillator An oscillator resistor is connected between OSC and VSS
VDD	I	—	Positive power supply
VSS	I	—	Negative power supply (GND)

Absolute Maximum Ratings*

Supply Voltage -0.3V to 6V Storage Temperature..... -50°C to 125°C
Input Voltage..... VSS-0.3V to VDD+0.3V Operating Temperature..... -20°C to 70°C

*Note: These are stress ratings only. Stresses exceeding the range specified under “Absolute Maximum Ratings” may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

Electrical Characteristics

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
VDD	Operating Voltage	—	—	2.4	—	5.2	V
ISTB	Standby Current	3V	No load, system HALT	—	1	3	μA
IDD	Operating Current	3V	No load, fSYS=4MHz	—	5	8	mA
IOL	PB (0:7) Sink Current	3V	VOL=0.3V	4	6	—	mA
IO	Max. AUD1 and AUD2 Output Current	3V	VOH=0.6V	-1.5	-2	—	mA
fSYS	System Frequency	3V	ROSC=180kΩ	3.6	4.0	4.4	MHz

Functional Description

The Magic Voice™ series is a series of programmable speech synthesizers and melody generators. It provides various sample rates and beats, tone levels, tempos for speech synthesizer and melody generator. For voice synthesizer, the Magic Voice™ provides 8-bit PCM, 6-bit μ -law PCM and 4-bit ADPCM synthesis. In HT84XXX series, a mixed output of two channels PCM synthesis is allowed but only one channel ADPCM synthesis is allowed. For melody generator, the Magic Voice™ supports a tone melody and PCM melody generator and two channels mixed output for the whole series.

The Magic Voice™ series build-in 8 programmable I/O pins and 4 programmable input pins along with powerful user's command. The user's instructions are employed to develop new and customized functions for a wide variety of innovative applications.

Speech and melody analysis

The speech and melody sources of the Magic Voice™ can be recorded and edited from the PC sound card and media tools. HOLTEK's CAD tools first load a speech source file as .WAV or .PCM format, then transfer the speech file as PCM, LOG-PCM or ADPCM format, and finally save it to the internal mask ROM by changing a layer of the mask. The PCM format generates a higher sound quality whereas the ADPCM format brings about a longer recording capacity. The melody source can either be in the .MID or in the .MLD format.

The .MID file is the standard format of the windows media tools. The .MLD file is a text format. After the .MID file is compiled, the .MLD file is automatically generated.

The Magic Voice™ can support the following compression format of the voice .WAV and .PCM file: AD4, PCM8, μ PCM.

Current type D/A output

The HT84XXX series supply two high accuracy current type D/A output pins for audio output. The output volume is changeable from 0 to 15 digital levels by writing a value to the VOL_UME_n (n=1 or 2) registers. The D/A pins are PMOS open drain structure and output synthesized signals for driving a speaker through an external NPN transistor when the chip is active. However, it becomes floating when the chip is in the standby state. An 8050 type transistor with $h_{FE}=150$ is recommended for the output driver of the D/A output pin.

Melody/Tone generator

The HT84XXX family has a built-in melody/tone generator. The generator can generate 13 different kinds of melody beats, 5 octaves of tone level, 18 tempos, and 2 channels mixed output. Of these components, the melody tempo is changeable and generates sound effects by writing a control value to the TEMPO register. The chip provides the following 18 tempos, 13 beats, and 5 octaves for user's programming.

• 18 tempos (Beats/Min.)

68	78	82	93	100	105
109	114	119	125	132	139
147	156	179	192	227	310

• 13 beats

$\frac{1}{24}, \frac{1}{12}, \frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{3}{4}, \frac{1}{3}, \frac{4}{3}, \frac{2}{3}, \frac{3}{2}, \frac{4}{3}, \frac{1}{3}, \frac{2}{3}$

• 5 octaves

C1~B1, C2~B2, C3~B3, C4~B4, C5~B5

PCM/ADPCM synthesizer

The HT84XXX family contains a PCM and ADPCM synthesizer. The synthesizer offers a wide range of sampling rates from 4kHz to 24kHz for PCM synthesis and 4kHz to 16kHz for ADPCM synthesis. The sample rate of the synthesizer can be changed by writing a control value to the sample rate register. It also supports a small variety of 100Hz. For a higher performance sound quality, the PCM coding is required. But for a longer recording capacity, the ADPCM coding is recommended.

The capability of the voice sample rate for various voice compression format.

Fosc	PCM8	μPCM	AD4
4M	12kHz	11kHz	8kHz
5M	15kHz	14kHz	10kHz
6M	18kHz	16kHz	12kHz
8M	24kHz	22kHz	16kHz

Oscillator configuration

The HT84XXX series provides an RC oscillator for the system clock. The system oscillator stops in the standby state so as to reduce power consumption.

For the oscillator circuit, an external resistor is required between OSC and VSS. The oscillator frequency is typically 4MHz for an external resistor of 180kΩ. The RC type of oscillator offers the most cost-effective solution, although the frequency of the oscillation may vary with temperature and the chip itself due to process variation.

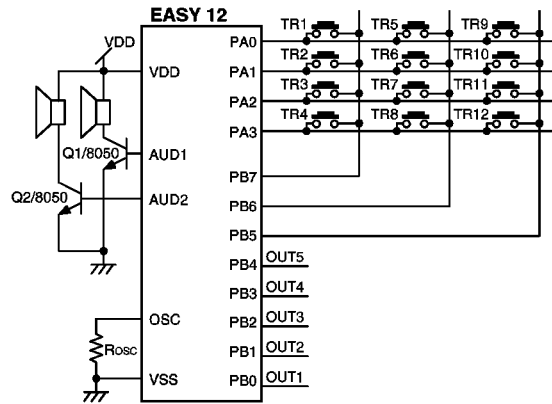
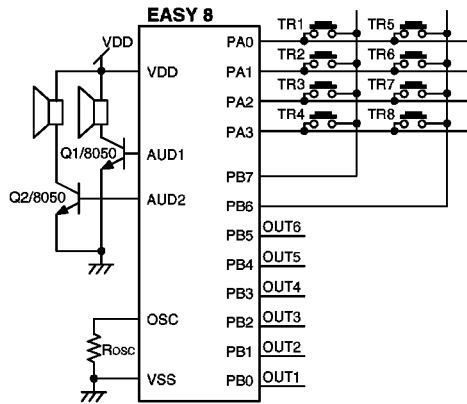
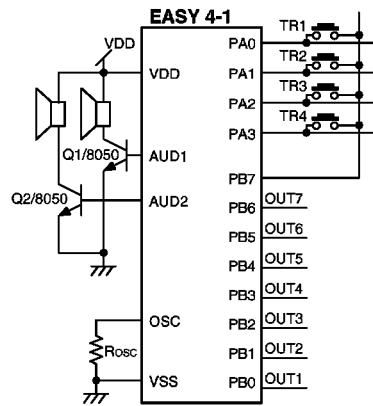
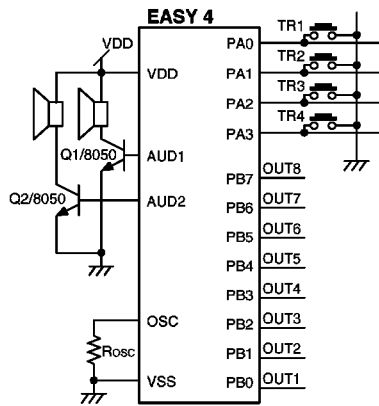
Fosc	EV R _{osc}	IC R _{osc}
4M	180K	240K
5M	150K	2210K
6M	120K	180K
8M	91K	150K

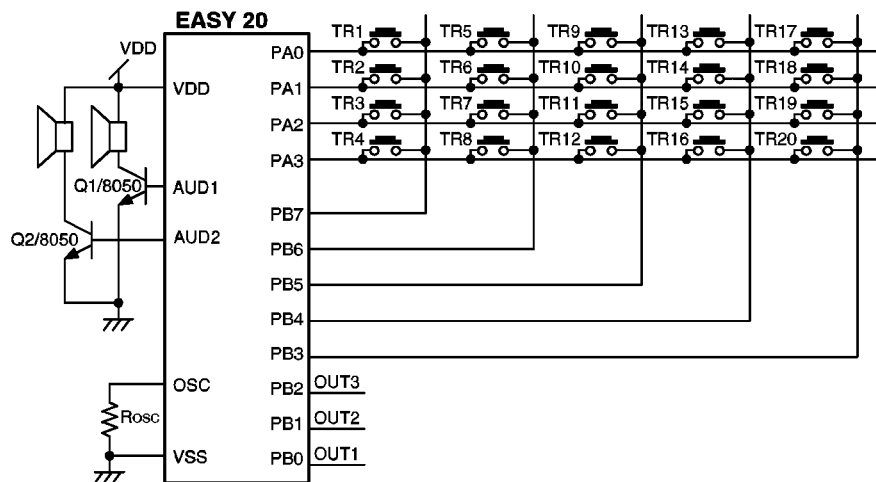
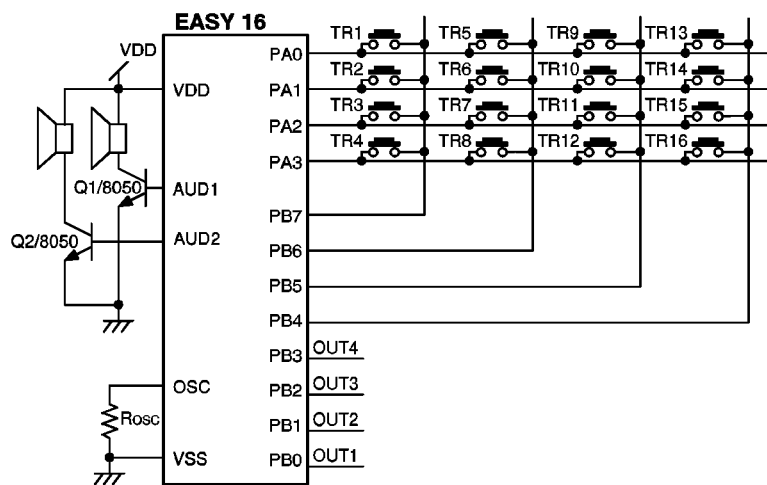
Mask options

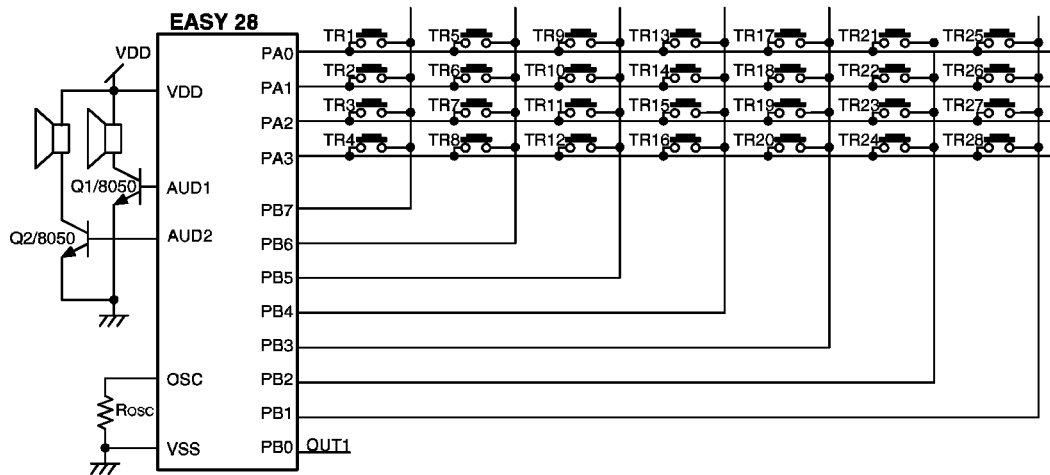
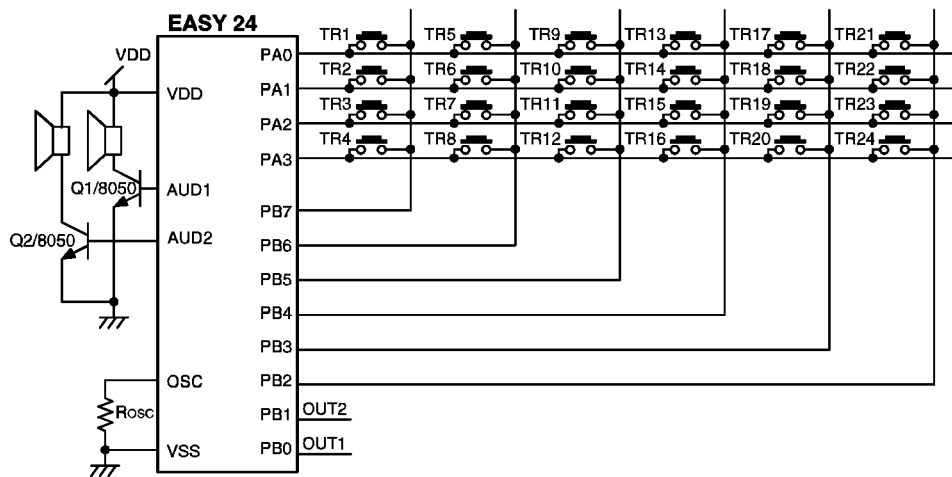
The following options have to be defined to ensure a proper system functioning:

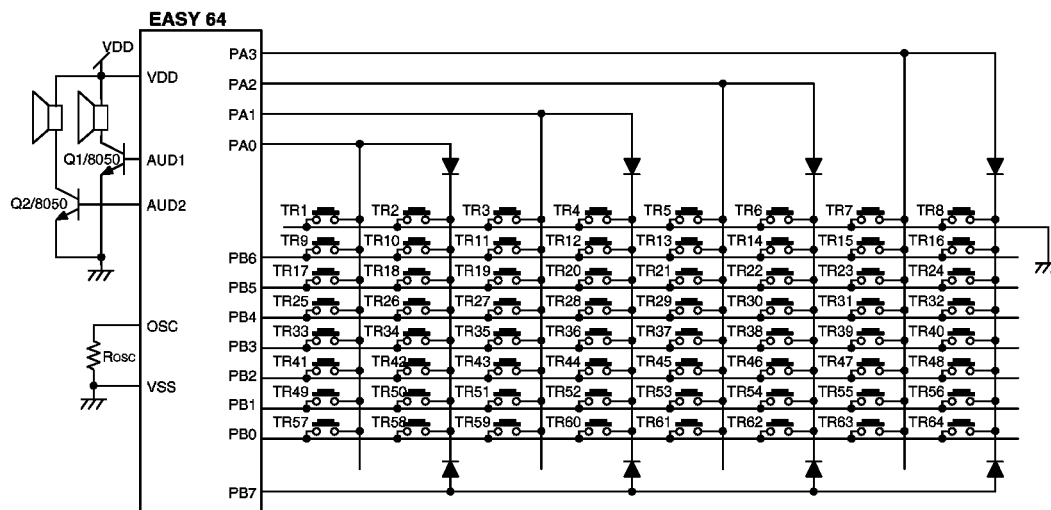
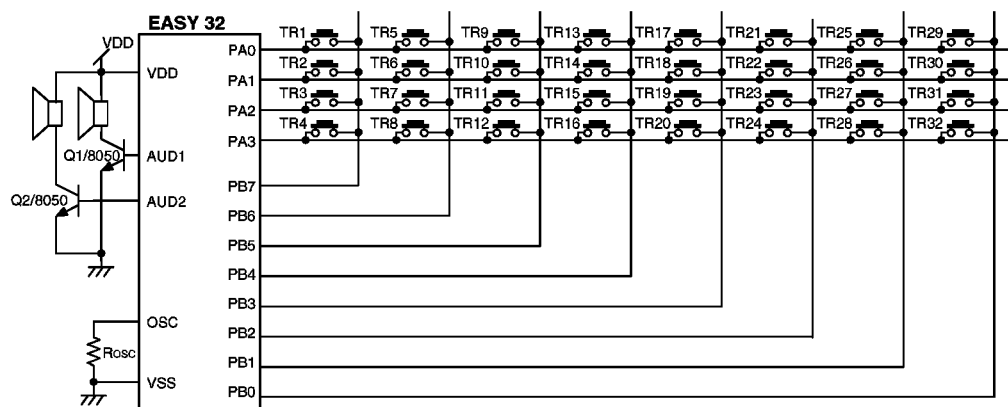
- Pull-high resistor: 33K/98K
- Key debounce time: 0ms~255ms

Application Circuits (BASIC)



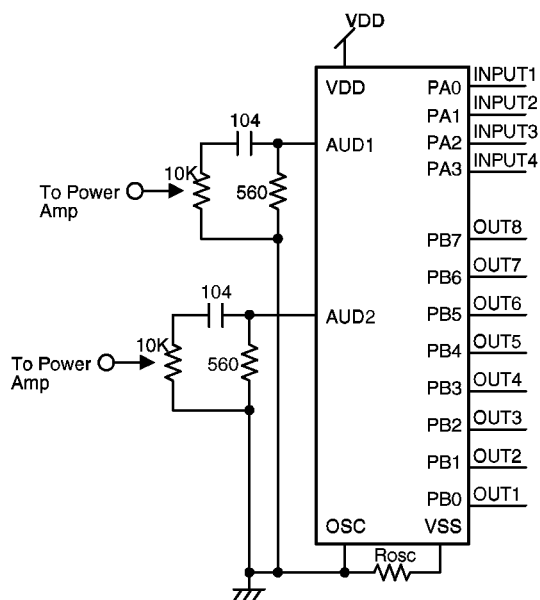
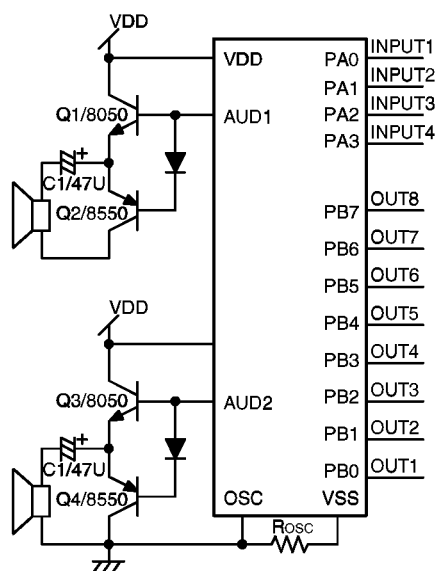






Notes: In Easy 64 mode, only falling edge trigger can be used in input state

In Easy 64 mode, the path command "TRn?L:pathname" and "TRn?H:pathname" are invalid



Application Notes

Easy Format™

The syntax of the Easy Format™ consists of four major parts. They are format and audio file declaration part, input states declaration part, output states declaration part and path command part. The architecture of the Easy Format™ program is shown as follows:

```

EASY n          ; Comment
                ; Format and audio file declaration part
....
Stereo/Mono     ; Two AUD output or single AUD output
INPUT STATES    ; Input states declaration part
....
OUTPUT STATES   ; Output states declaration part
....
PATHS           ; Path command part
....

```

Note: Easy Format™ is a trademark of Grow With Me, Inc.

Format and voice/melody file declaration

In this area the format and audio file declaration are used to define the number of input and output as well as declare the audio format.

Syntax: EASY n

The EASY n command is used to define the maximum number of trigger input in the application circuit. The alphanumeric n represents the number of trigger input where n has the choice from 4 to 64 with a scale 4. However, the determination of trigger input reflects the number of output. The relation of the number of I/O is shown in the following table.

EASY n	TR n	OUT n
EASY 4	TR1, TR2, TR3, TR4	OUT 1, OUT 2, ..., OUT 7
EASY 8	TR1, TR2, ..., TR8	OUT 1, OUT 2, ..., OUT 6
EASY 12	TR1, TR2, ..., TR12	OUT 1, OUT 2, ..., OUT 5
EASY 16	TR1, TR2, ..., TR16	OUT 1, OUT 2, ..., OUT 4
EASY 20	TR1, TR2, ..., TR20	OUT 1, OUT 2, OUT3
EASY 24	TR1, TR2, ..., TR24	OUT 1, OUT 2
EASY 28	TR1, TR2, ..., TR28	OUT1
EASY 32	TR1, TR2, ..., TR32	
EASY 64	TR1, TR2, ..., TR64	
* refer to the application CKT		

Syntax:

```
voice_file/compression_method
melody_file
tone_file
```

The source audio files must be included in your Easy program. The audio files include the voice files with the extended name .PCM or .WAV, the melody files with .MID or .MLD format and the tone files with HOLTEK's .HT8 format. The compression methods of voice files have three options such as μ PCM, PCM8 and AD4.

Example:

```
voice1.wav/pcm8      ; #0 voice files declaration
voice2.pcm/ad4       ; #1
melody1.mid          ; #2 melody files declaration
melody2.mld          ; #3
tone1.ht8            ; #4 tone files declaration
```

Stereo/Mono:

The audio output selection. If the audio output is declared as "Stereo", the channel 1 voice will output via AUD1 and the channel 2 voice will output via AUD2. If the audio output is declared as "Mono", the channel 1 and channel 2 voices will mixedly output via AUD1 and AUD2 is invalid. The default declaration is "Mono". The audio file compressed by AD4 cannot output through channel 2. If the melody file outputs through channel 2, the melody file cannot include any channel 1 instructions.

Input states declaration

Syntax:

```
INPUT STATES
;          TRn          ....
statename: [rising_edge_path]/[falling_edge_path]  ....
```

The bracket [] is denoted as optional existence. If the path is assigned as "X", the input trigger signal is ignored. Unassigned trigger paths will automatically be assigned as "X". When TRn detects a rising edge or a falling edge signal, the path name of the rising edge path or falling edge path will be executed. The statename and path label can be any name defined by the user. The "/" denotes the falling edge path. The maximum number of statename are 256 and the maximum number of different pathname are 128.

Example:

```
INPUT STATES  ; input states declaration
;            TR1          TR2          TR3
Run:         P12Start     /P13Stop     P14Go/P15Pause
```

where "Run" denotes the input statename and "P12Start", "/P13Stop", "P14Go" and "/P15Pause" are all pathnames. When a rising edge signal triggers the TR1, then the path of P12Start will be executed. And if a falling edge signal triggers the TR1, then the signal is ignored, etc.

Output states declaration

Syntax:

```

OUTPUT STATES
;          OUT n      ....
statename: OutputMode ....

```

The possible output mode are listed and described on the following table. The maximum number of output state are 256.

OutputMode	Description
X	Set OUTn to input mode with pull high resistor
H	Set the output pins to logic high
L	Set the output pins to logic low
P+	Send a (+) pulse train to output pins
P-	Send a (-) pulse train to output pins
SP+	Send a plus single pulse to output pins
SP-	Send a minus single pulse to output pins

Example:

```

OUTPUT STATES          ; output states declaration
;          OUT 1      OUT 2
SirenOn:      P+      L

```

where “SirenOn” denotes the output statename. Executing “SirenOn” in the path command will send a positive pulse train to PB0 and send a logic low to PB1.

Path command definition

Syntax:

```

PATHS
pathname:  path command  ....

```

The pathname is defined according to the user’s desire. The elements of the path command are listed and described on the following table. Each path equation consists of many path commands and each command is separated by one or more space. In order to increase the program readability if the path commands exceed the screen display you can break the path command to the next line and put the symbol “&” in front of the new line.

Path_Command	Description
Input statename	To active input state
Output statename	To active output state
Pathname	To execute the specified path
VAR=pathname	Define the variable path
VAR	To execute the variable path
Volume_1=n	Define the volume 1 value (n=0~15)
Volume_2=n	Define the volume 2 value (n=0~15)
Samplerate_1=n	Define the sample rate 1 value
Samplerate_2=n	Define the sample rate 2 value
Delay(n)	To delay n (min=0.1s) (n=0.1~65)
TRn?H:pathname	If TRn is logic High, then the pathname is executed.
TRn?L:pathname	If TRn is logic Low, then the pathname is executed.
Flashrate=n	Set the output pulse rate to “n” pulses per second, where n can be assigned as 1, 2, 3, ..., 12
FlashrateT=n	Set the output pulse rate after the melody tempo, n=1, 2, 3, ..., 12
FlashrateV=n	Set the output pulse rate after the volume, n=1, 2, ..., 12
END	Enter power down mode
Mi=PB	Read Port B to Mi
STOP1	Stop the sound of channel 1 playback
STOP2	Stop the sound of channel 2 playback
Mi=data	Set the contents of Mi to be data
Mi=Mj	Set the contents Mi to be Mj
Mi(bn)=1 or 0	Set bit bn to be 1 or 0, bn=0~7
Mi=Mj+data	Add immediate data with Mj to Mi
Mi=Mi+Mj	Add Mi with Mj to Mi
Mi=Mj.AND.data	And immediate data with Mj to Mi
Mi=Mj.OR.data	OR immediate data with Mj to Mi
Mi=Mj.XOR.data	XOR immediate data with Mj to Mi
Mi?data:pathname	If Mi=data then pathname is executed
Mi?Mj:pathname	If Mi=Mj then pathname is executed.
Mi(bn)?1:pathname	If the bit bn of Mi is “1” then the pathname is executed.
Mi(bn)?0:pathname	If the bit bn of Mi is “0” then the pathname is executed.

Path_Command	Description
Mi:[Path1, Path2, Path3, ..., Pathn]	Path1 is executed when Mi=1; Path2 is executed when Mi=2; ... Pathn is executed when Mi=n
Random(Mi)	Get a random code and put it to Mi
Timert: timer_path	Initial the timer. If time out then timer_path is executed, t=0.1~65 sec.
TimerON	Start the timer counter
TimerOFF	Stop timer counter
n*soundfile, n*#N or #N	Play the soundfile n times #N is the N'th soundfile which is defined in the audio file declaration part. The starting number is 0
[n*soundfile] or [repeat *soundfile]	Play the soundfile n times or repeatedly via channel 2 and simultaneously execute the next path command.
Wait	Stop executing the next path command until the channel 2 soundfile is terminated.

Notes: The n of TRn ranges from 1 to 32

“Volume_1=15” is equal to “Volume=15”

“Samplerate_1=4000” is equal to “Samplerate=4000”, “4000” means that the sample rate is 4kHz.

The Mi, Mj are working registers (i, j=0, 1, 2, ..., 10), the M0~M4 are dedicated for users, the others are shared with VAR and Timer command.

Register	Command	Comment
M0~M4	—	Normal register
M5, M6	VAR	If the VAR instruction is used, the M5, M6 will be invalid.
M7~M10	Timer	If the Timer instruction are used, the M7~M10 will be invalid.

Example:

```

EASY 4
INPUT STATES
;          TR1    TR2    TR3    TR4
start:    /P1     X      X      X

PATHS
POWERON:  start          END

```

where “POWERON” is a reserved pathname and provides the initial setting. When the power is turned on, a falling edge trigger in TR1 is accepted and others are all ignored.

Application 1 — One shot, nonretriggerable

EASY 4

voice.wav/pcm8 ;#0 sound file declaration

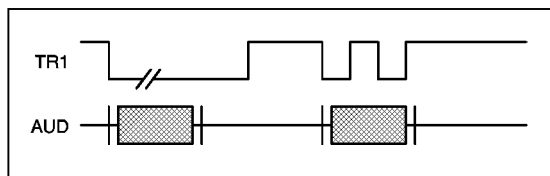
INPUT STATES

	TR1	TR2	TR3	TR4
state1:	/path1	X	X	X
busy:	X	X	X	X

PATHS

POWERON: state1 END

path1: busy 1*voice state1 END



Application 2 — One shot, retriggerable by itself

EASY 4

voice.wav/pcm8 ;#0 sound file declaration

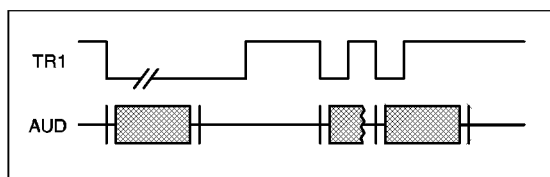
INPUT STATES

	TR1	TR2	TR3	TR4
state1:	/path1	X	X	X

PATHS

POWERON: state1 END

path1: 1*voice END

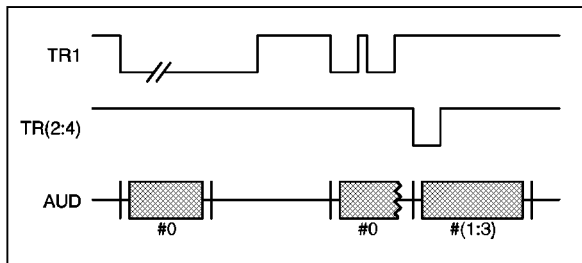


EASY 4

```
voice4.wav/pcm8 ;#3
```

	TR1	TR2	TR3	TR4
state0:	/path1	/path2	/path3	/path4
state1:	X	/path2	/path3	/path4
state2:	/path1	X	/path3	/path4
state3:	/path1	/path2	X	/path4
state4:	/path1	/path2	/path3	X

POWERON:	state0	END		
path1:	state1	#0	state0	END
path2:	state2	#1	state0	END
path3:	state3	#2	state0	END
path4:	state4	#3	state0	END



Application 4 — Level hold, retriggerable by the other pin only

EASY 4

voice1.wav/pcm8 ;#0 sound file declaration

voice2.wav/pcm8 ;#1

voice3.wav/pcm8 ;#2

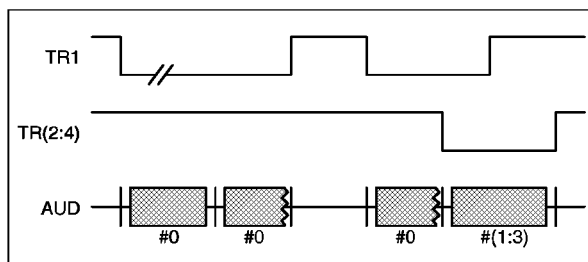
voice4.wav/pcm8 ;#3

INPUT STATES

	TR1	TR2	TR3	TR4
state0:	/path1	/path2	/path3	/path4
state1:	path11	/path2	/path3	/path4
state2:	/path1	path11	/path3	/path4
state3:	/path1	/path2	path11	/path4
state4:	/path1	/path2	/path3	path11

PATHS

POWERON:	state0	END	
path1:	state1	#0	path1
path2:	state2	#1	path2
path3:	state3	#2	path3
path4:	state4	#3	path4
path11:	state0	END	



Application 5 — LED output, direct mode

EASY 4

voice1 .wav/pcm8 ;#0 sound file declaration

INPUT STATES

```

;          TR1      TR2      TR3      TR4
state0:    /path1    X        X        X

```

OUTPUT STATES

```

;OUT       1        2
alarm:     P+       L
standby:   L        H
turnoff:   X        X

```

PATHS

```

POWERON:   state0    flashrate=6Hz      END
path1:     alarm     #0      standby    delay(2)    turnoff    END

```

Application 6 — LED output matrix mode

EASY 4

INPUT STATES

```

;          TR1
state0:    /path0

```

OUTPUT STATES

```

;          OUT1  OUT2  OUT3  OUT4  OUT5  OUT6  OUT7
LED1:     P+    L     L     L     H     H     H
LED2:     L     P+    L     L     H     H     H
LED3:     L     L     P+    L     H     H     H
LED4:     P+    L     L     H     L     H     H
LED5:     L     P+    L     H     L     H     H
LED6:     L     L     P+    H     L     H     H
LED7:     P+    L     L     H     H     L     H
LED8:     L     P+    L     H     H     L     H
LED9:     L     L     P+    H     H     L     H
LED10:    P+    L     L     H     H     H     L
LED11:    L     P+    L     H     H     H     L
LED12:    L     L     P+    H     H     H     L
turnoff:   X     X     X     X     X     X     X

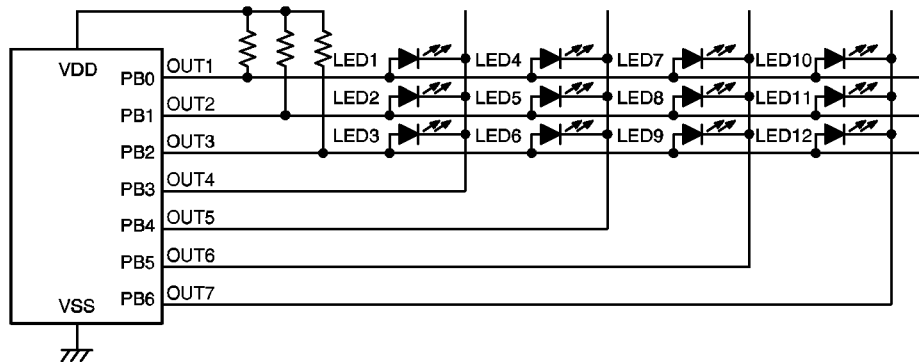
```

PATHS

```

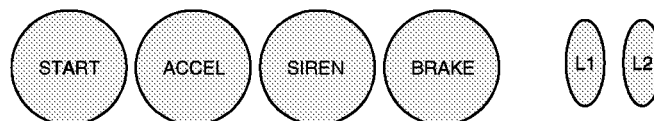
POWERON:      state0 M0=0  flashrate=3Hz  END
path0:        M0=M0+1
               & M0:[P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12]

P1:           LED1  delay(2)  turnoff  END
P2:           LED2  delay(2)  turnoff  END
P3:           LED3  delay(2)  turnoff  END
P4:           LED4  delay(2)  turnoff  END
P5:           LED5  delay(2)  turnoff  END
P6:           LED6  delay(2)  turnoff  END
P7:           LED7  delay(2)  turnoff  END
P8:           LED8  delay(2)  turnoff  END
P9:           LED9  delay(2)  turnoff  END
P10:          LED10 delay(2)  turnoff  END
P11:          LED11 delay(2)  turnoff  END
P12:          M0=0 LED12 delay(2) turnoff  END
    
```



Application 7 — Police car

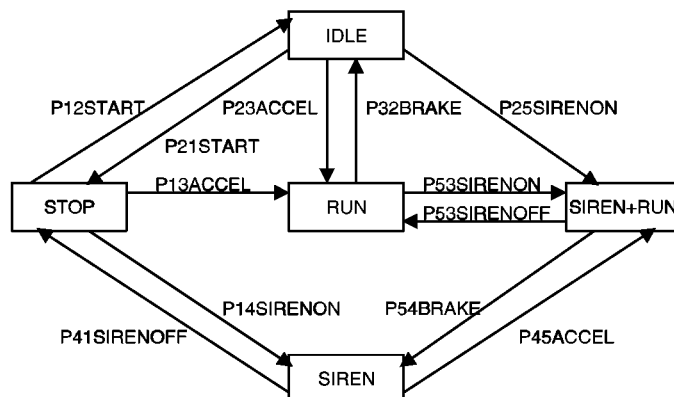
This is a full program application example to demonstrate how Easy Format handles a complicated design requirement. All the work is to be done on a single page.



There are four input trigger buttons and two output pins to drive two LEDs.

- BUTTON #1: Turn ON (START) or OFF the engine.
- BUTTON #2: Gas pedal to give ACCELERATION sound from idle and then keep on at a steady running sound.
- BUTTON #3: SIREN sound On/Off: toggle ON and toggle Off.
- BUTTON #4: BRAKE, triggers deceleration sound and then brings the car to idle.
- LEDs OUTPUTs: Blinks in opposite phase when the Siren sound is on. That is when one light is on, the other light is off in an alternating pattern.
- TIME OUT REQUIREMENTS: every sound needs to be automatically shut down after a specified period if no more trigger signal is received.

Easy Format™ application example: Police Car flow chart



EASY 4

;* voice file define

StartSnd.wav/ad4

IdleSnd.wav/ad4

AccelSnd.wav/ad4

RunSnd.wav/ad4

DecelSnd.wav/ad4

SirenSnd.wav/ad4

SiRunSnd.wav/ad4

;* I/O States

INPUT STATES

;	TR1	TR2	TR3	TR4
;	Start	Accel	Siren	Brake
StopCar:	/P12Start	/P13Accel	/P14SirenOn	X
Idle:	/P21Start	/P23Accel	/P25SirenOn	X
Run:	X	X	/P35SirenOn	/P32Brake
Siren	X	/P45Accel	/P41SirenOff	X
SirenRun	X	X	/P53SirenOff	/P54Brake

OUTPUT STATES

;	OUT1	OUT2
SirenOn:	P+	P-
SirenOff:	H	H

;* Paths

Paths

POWERON:	StopCar	SirenOff	FLASHRATE=6Hz	END
P12Start:	Idle	1*StartSnd	10*IdleSnd	PowerDown
P21Start:	StopCar	END		
P13Accel:	Run	2*IdleSnd	1*AccelSnd	20*RunSnd
		& PowerDown		
P14SirenOn:	Siren	SirenOn	10*SirenSnd	PowerDown


```

P41SirenOff:    StopCar  SirenOff  END
P23Accel:       Run   1*AccelSnd  20*RunSnd  PowerDown
P32Brake:       Idle  1*DecelSnd  10*idleSnd  PowerDown
P25SirenOn:     1*AccelSnd  SirenRun  SirenOn  10*SiRunSnd  PowerDown
P35SirenOn:     SirenRun  SirenOn  10*SiRunSnd  PowerDown
P53SirenOff:    SirenOff  Run   20*RunSnd  PowerDown
P45Accel:       P25SirenOn
P54Brake:       1*DecelSnd  Siren  SirenOn  10*SirenSnd
                & PowerDown
PowerDown:      SirenOff  StopCar  End

```