



HT84XXX Magic Voice™

Features

- Operating voltage: 2.4V~5.0V
- Programmable tone melody generator
- ADPCM or μ -law PCM, PCM synthesis
- Wide range of sampling rate for voice synthesis
- Minimum sampling 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 36 sec to 384 sec of voice capacity
- Timer controller
- Voice fill-in function
- Powerful user-defined functions
- Power-on initial setting
- 28-pin SKDIP package

Applications

- High-end educational leisure products
- 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 sampling rates and beats, tone levels, tempos for the speech synthesizer and melody generator. The HT84XXX series has two built-in 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.

The HT84XXX Magic Voice™ is Easy Format™ supported.

ROM Selection Table

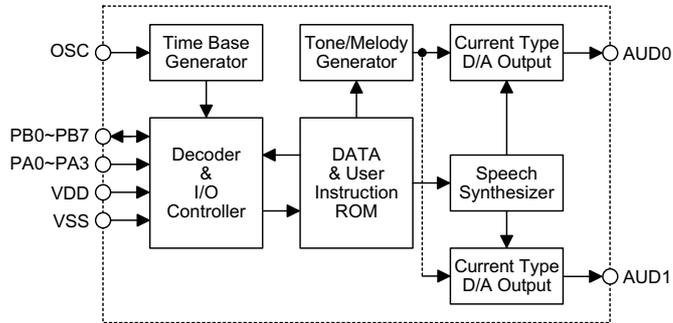
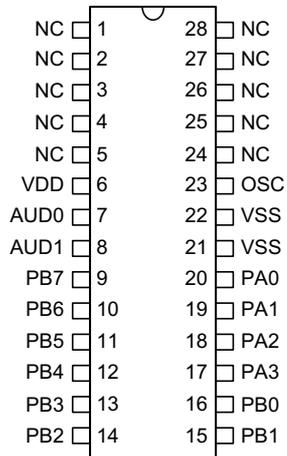
The HT84XXX series provides various voice capacity as shown below:

Part No.	HT84036	HT84072	HT84144	HT84192	HT84384
ROM	768Kb	1536Kb	3072Kb	4096Kb	8192Kb
Voice Length	36 sec	72 sec	144 sec	192 sec	384 sec

Note: The voice capacity is based on a sampling rate of 21Kb/s

Magic Voice™ is a trademark of Holtek Semiconductor Inc.
Easy Format™ is a trademark of Grow With Me, Inc.

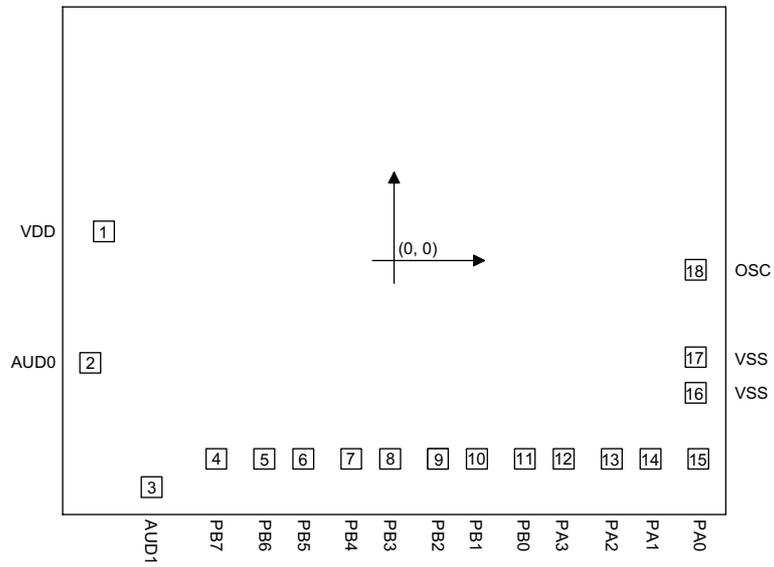


Block Diagram

Pin Assignment


**HT84XXX
-28 SKDIP**

Pad Assignment

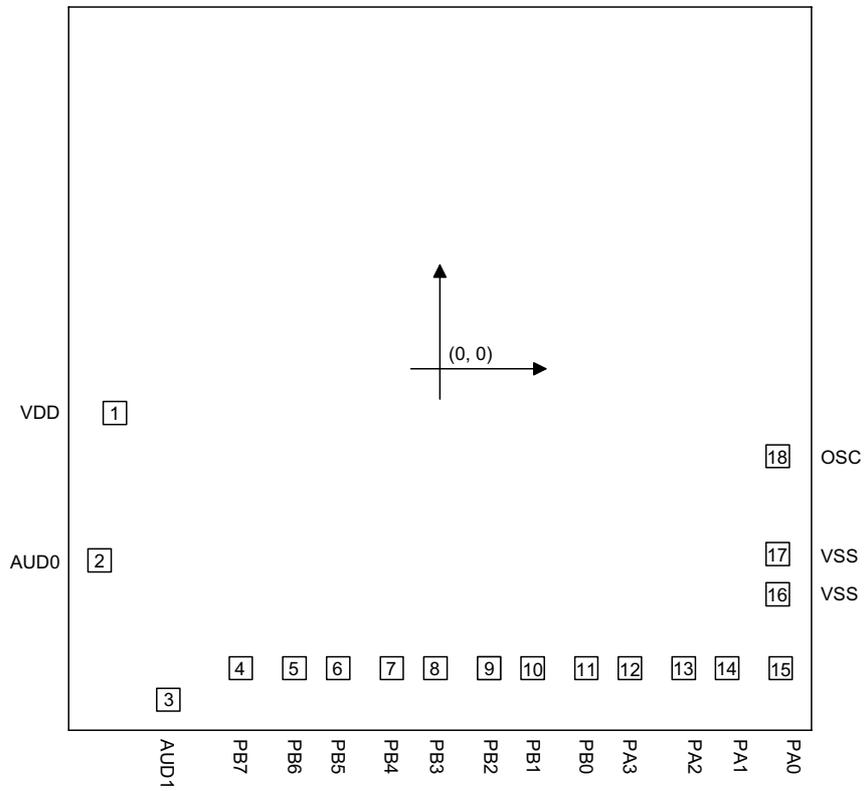
HT84036



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

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

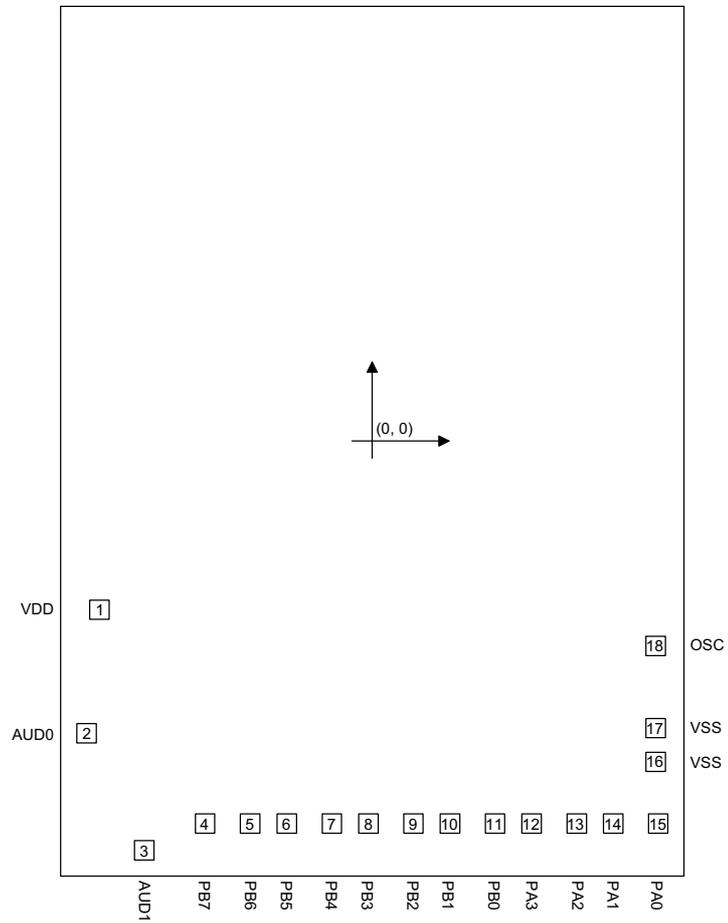
HT84072



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

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

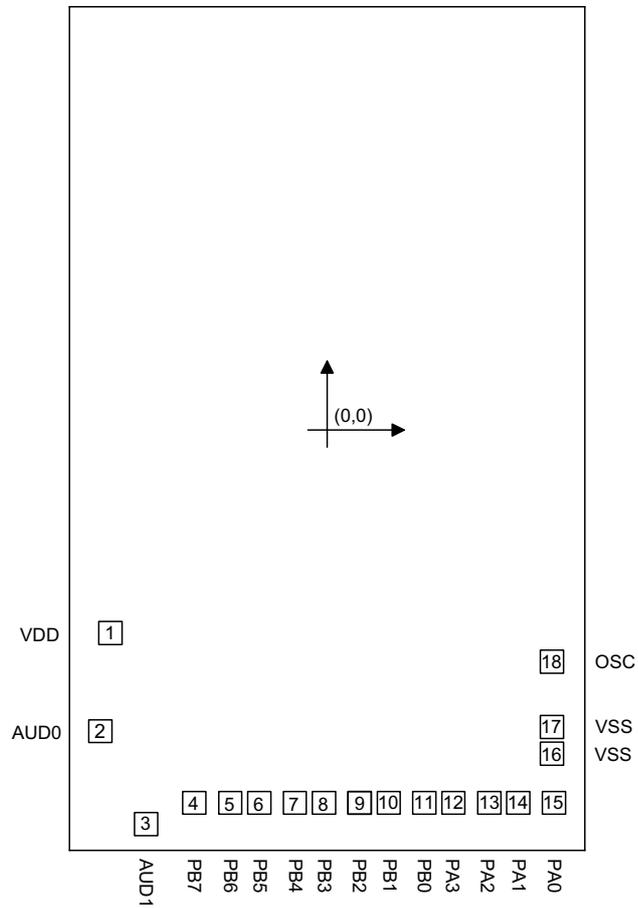
HT84144



Chip size: $2735 \times 3735 (\mu\text{m})^2$

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

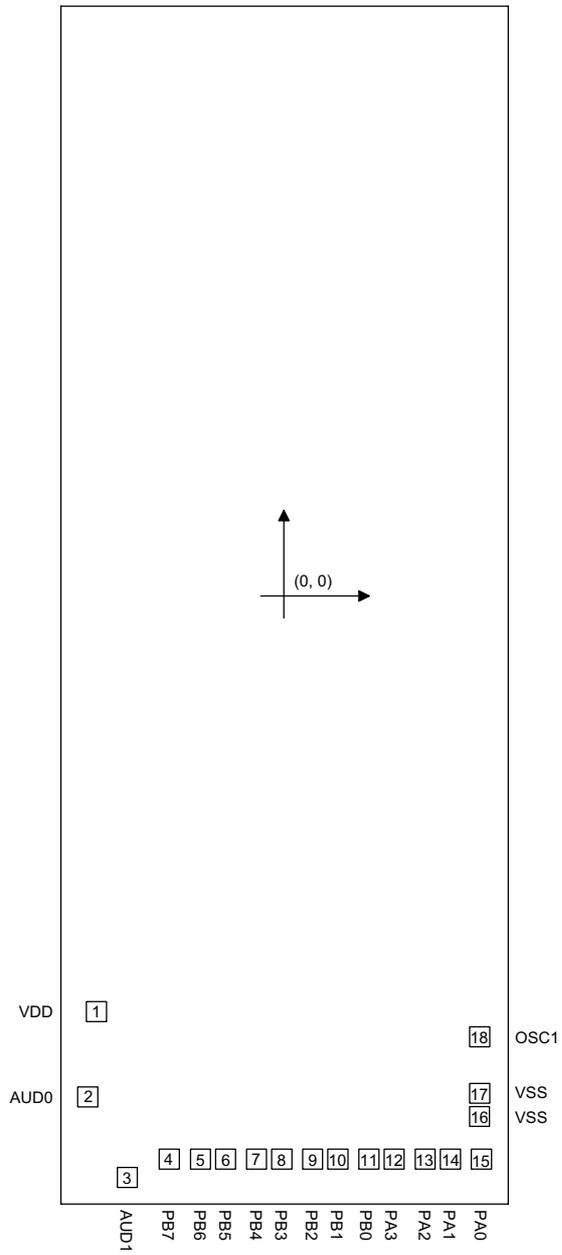
HT84192



Chip size: $2735 \times 4460 (\mu\text{m})^2$

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

HT84384



Chip size: 2740 × 7350 (μm)²

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

Pad Coordinates
HT84036

 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

HT84072

 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

HT84144

 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

HT84192

 Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	-1125.31	-1062.76	10	320.15	-1951.74
2	-1177.49	-1576.29	11	505.75	-1951.74
3	-939.68	-2062.11	12	656.55	-1951.74
4	-689.05	-1951.74	13	842.15	-1951.74
5	-503.45	-1951.74	14	992.95	-1951.74
6	-352.65	-1951.74	15	1178.55	-1951.74
7	-167.05	-1951.74	16	1167.43	-1694.68
8	-16.25	-1951.74	17	1167.53	-1554.58
9	169.35	-1951.74	18	1167.53	-1213.46

HT84384

 Unit: μm

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

Pin Description

Pin No.	Pin Name	I/O	Internal Connection	Description
1~5, 24~28	NC	—	—	No connection
6	VDD	—	—	Positive power supply
7	AUD0	O	PMOS Open Drain	Audio output for driving an external transistor
8	AUD1	O	PMOS Open Drain	Audio output for driving an external transistor
15~9	PB1~PB7	I/O	Pull-high or CMOS	Bidirectional I/O pins Can be optioned as trigger inputs or LED outputs
16	PB0	I/O	Pull-high or CMOS	Bidirectional I/O pins Can be optioned as CDS interface with internal Schmitt trigger input

Pin No.	Pin Name	I/O	Internal Connection	Description
20~17	PA0~PA3	I	Wake-up Pull-high	Trigger inputs Can also be configured as wake-up inputs
21, 22	VSS	—	—	Negative power supply, ground
23	OSC	I	—	Built-in RC oscillator An oscillator resistor is connected between OSC and VSS

Absolute Maximum Ratings

Supply Voltage-0.3V to 6V Storage Temperature.....-50°C to 125°C
 Input Voltage $V_{SS}-0.3V$ to $V_{DD}+0.3V$ 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

 $T_a=25^\circ\text{C}$

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	—	5.2	V
I_{STB}	Standby Current	3V	No load, system HALT	—	1	3	μA
I_{DD}	Operating Current	3V	No load, $f_{SYS}=4\text{MHz}$	—	5	8	mA
I_{OL}	PB0~PB7 Sink Current	3V	$V_{OL}=0.3V$	4	6	—	mA
I_O	Max. AUD1 and AUD2 Output Current	3V	$V_{OH}=0.6V$	-1.5	-2	—	mA
f_{SYS}	System Frequency	3V	$R_{OSC}=240k\Omega$	3.6	4.0	4.4	MHz

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

f_{osc}	PCM8	μ -law PCM	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 re-

sistor of 240k Ω . 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.

f_{osc} \ R_{osc}	HT84EVA	HT84P00	IC
4M	180k Ω	240k Ω	240k Ω
5M	150k Ω	210k Ω	210k Ω
6M	120k Ω	180k Ω	180k Ω
8M	91k Ω	150k Ω	150k Ω

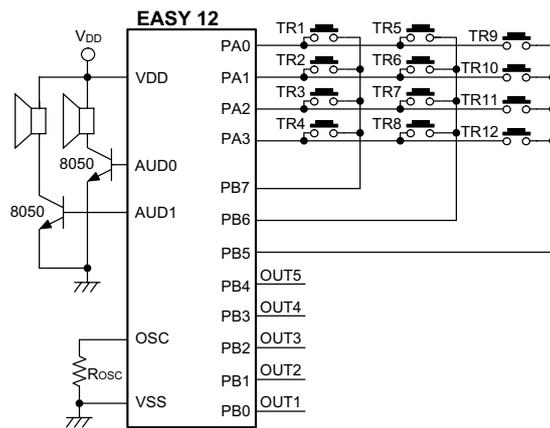
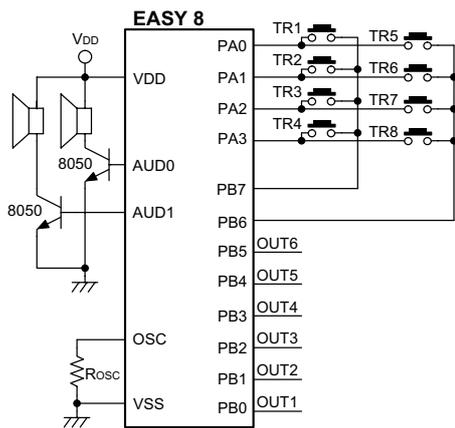
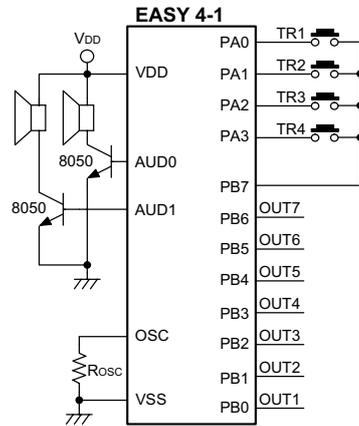
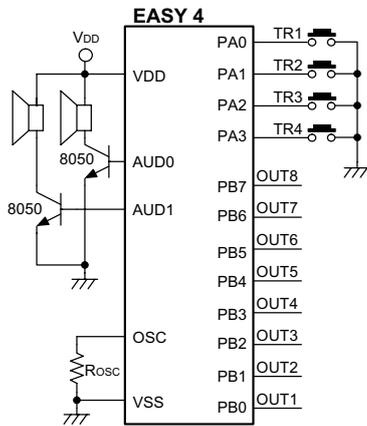
Mask options

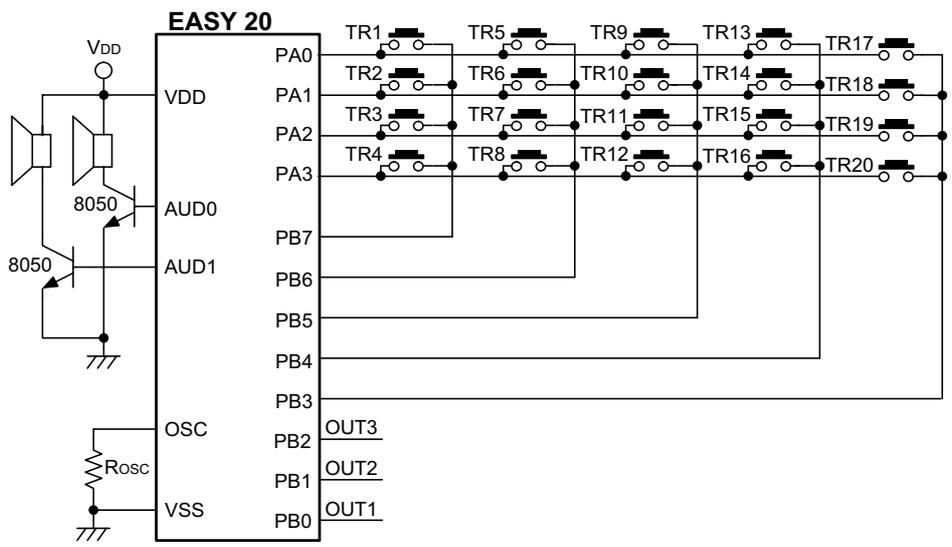
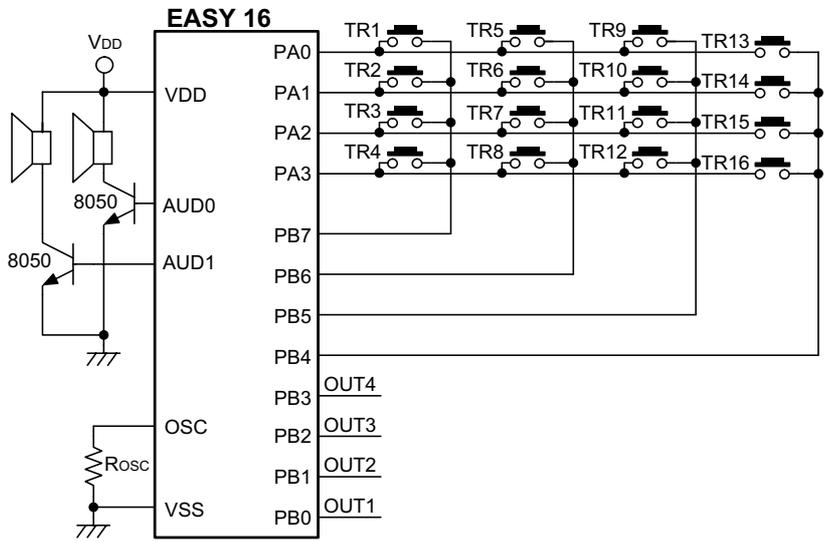
The following options have to be defined to ensure a proper system functioning:

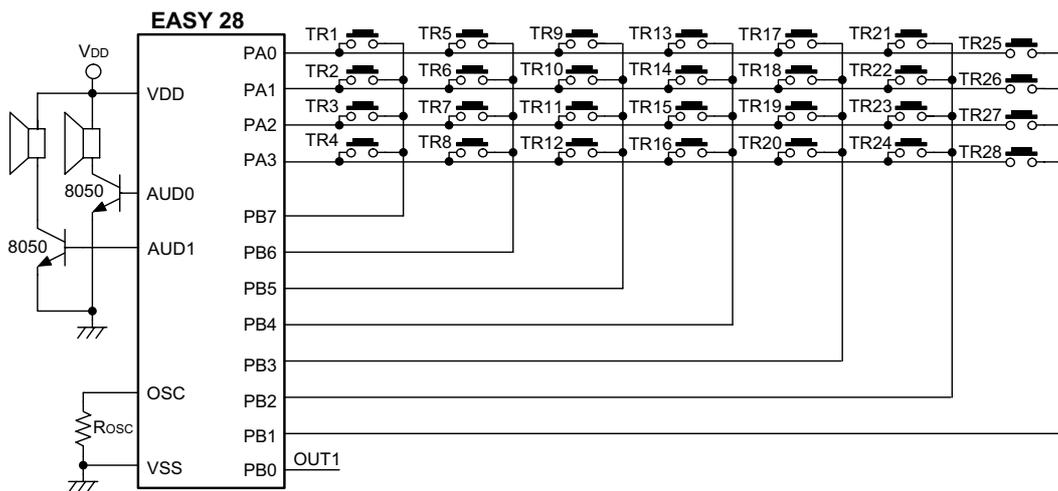
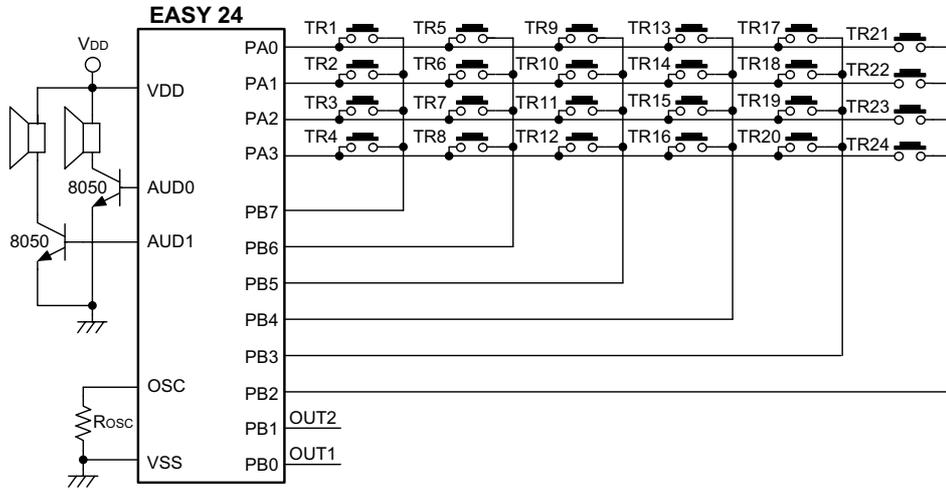
- Pull-high resistor: 33k Ω /98k Ω (3V)
- Keydebouncetime: 0ms~255ms (f_{osc} =4MHz)

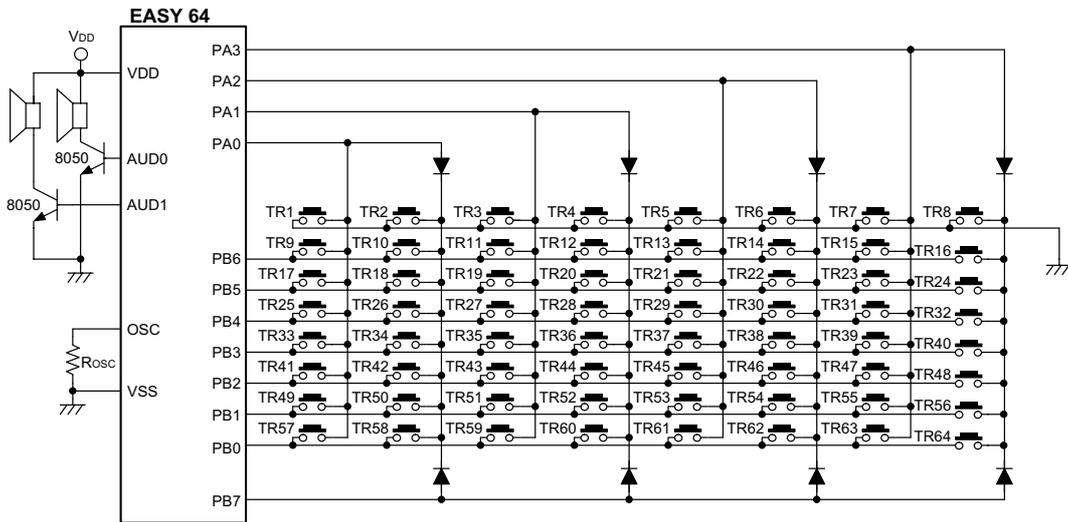
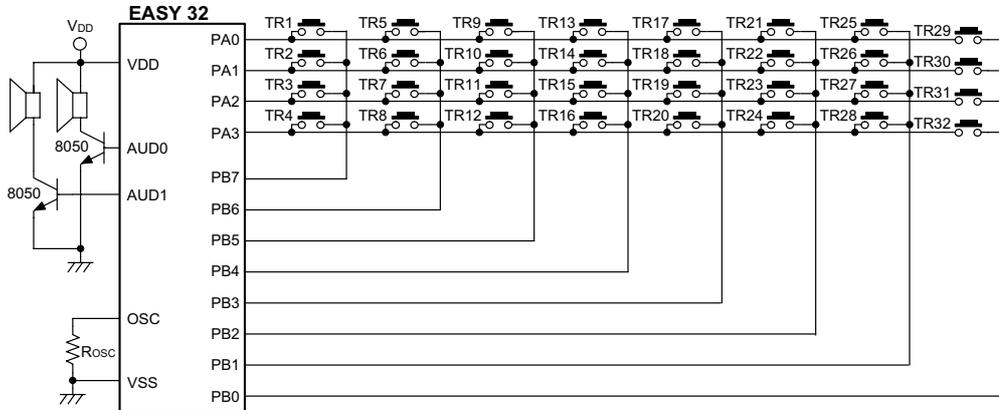
Application Circuits

Basic application





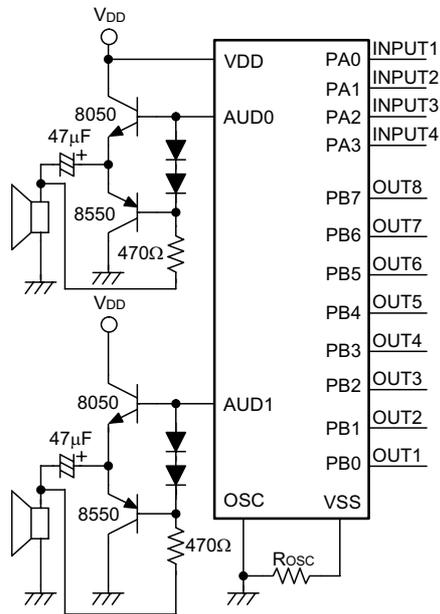




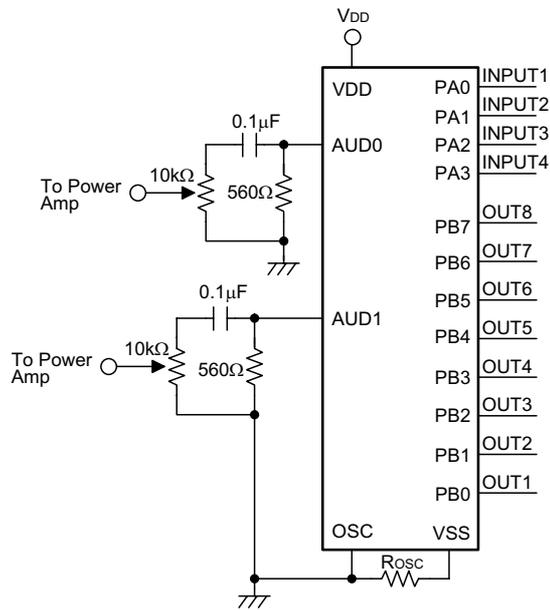
Note: In Easy 64 mode, only falling edge trigger can be used in input state

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

Push-pull amplifier application



OP amplifier application



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	TRn	OUTn
EASY 4	TR1, TR2, TR3, TR4	OUT1, OUT2, ..., OUT7
EASY 8	TR1, TR2, ..., TR8	OUT1, OUT2, ..., OUT6
EASY 12	TR1, TR2, ..., TR12	OUT1, OUT2, ..., OUT5
EASY 16	TR1, TR2, ..., TR16	OUT1, OUT2, ..., OUT4
EASY 20	TR1, TR2, ..., TR20	OUT1, OUT2, OUT3
EASY 24	TR1, TR2, ..., TR24	OUT1, OUT2
EASY 28	TR1, TR2, ..., TR28	OUT1
EASY 32	TR1, TR2, ..., TR32	
EASY 64	TR1, TR2, ..., TR64	
* refer to the application circuit		

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 -law 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
;          OUTn      ....
statename: OutputMode ....
    
```

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

Output Mode	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
;          OUT1      OUT2
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 sampling rate 1 value
Samplerate_2=n	Define the sampling rate 2 value
Delay(n)	To delay n (min=0.1s) (n=0.1~65), $f_{OSC}=4MHz$
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, data=0~255
Mi=Mj.OR.data	OR immediate data with Mj to Mi, data=0~255
Mi=Mj.XOR.data	XOR immediate data with Mj to Mi, data=0~255
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.

Note: 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 sampling 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

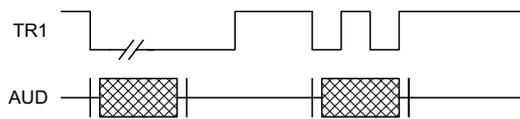
voice.wav/pcm8 ;#0 sound file declaration

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

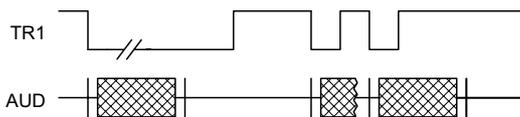
voice.wav/pcm8 ;#0 sound file declaration

INPUT STATES

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

PATHS

POWERON:	state1	END
path1:	1*voice	END



Application 3 — One shot, retriggerable by the other pin only

EASY 4

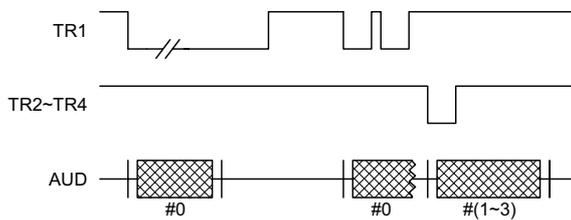
```
voice1.wav/pcm8 ;#0 sound file declaration
voice2.wav/pcm8 ;#1
voice3.wav/pcm8 ;#2
voice4.wav/pcm8 ;#3
```

INPUT STATES

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

PATHS

```
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

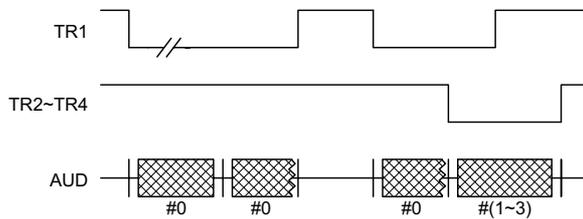
```
voice1.wav/pcm8 ;#0 sound file declaration
voice2.wav/pcm8 ;#1
voice3.wav/pcm8 ;#2
voice4.wav/pcm8 ;#3
```

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

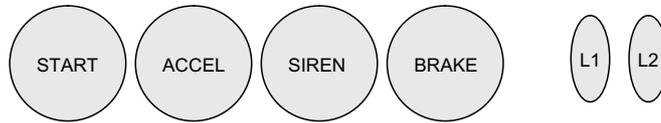
```

;          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

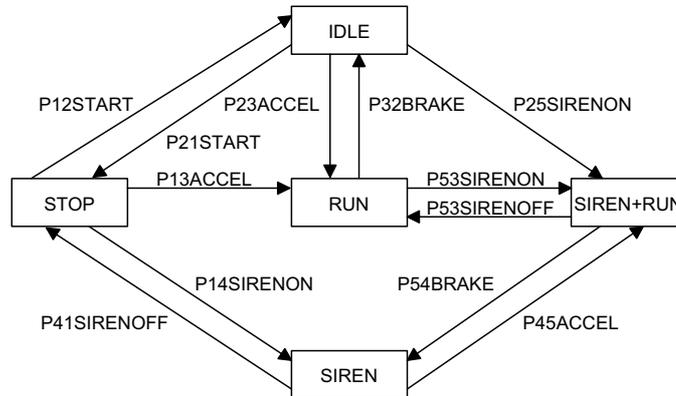
This is a full program application example to demonstrate how Easy Format handles a complicated design requirement. All the work is to be done on a single page.



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

Holtek Semiconductor Inc. (Headquarters)

No.3 Creation Rd. II, Science-based Industrial Park, Hsinchu, Taiwan, R.O.C.
Tel: 886-3-563-1999
Fax: 886-3-563-1189

Holtek Semiconductor Inc. (Taipei Office)

5F, No.576, Sec.7 Chung Hsiao E. Rd., Taipei, Taiwan, R.O.C.
Tel: 886-2-2782-9635
Fax: 886-2-2782-9636
Fax: 886-2-2782-7128 (International sales hotline)

Holtek Semiconductor (Hong Kong) Ltd.

RM.711, Tower 2, Cheung Sha Wan Plaza, 833 Cheung Sha Wan Rd., Kowloon, Hong Kong
Tel: 852-2-745-8288
Fax: 852-2-742-8657

Copyright © 2000 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at <http://www.holtek.com.tw>.