US 20080198139A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2008/0198139 A1

(43) Pub. Date: Aug. 21, 2008

Lacroix et al.

(54) HAPTIC FEEDBACK SYSTEM WITH STORED EFFECTS

(75) Inventors: **Robert Andre Lacroix**, Saint-Lambert (CA); **Pedro Gregorio**, Verdun (CA); **Kollin M. Tierling**, Milpitas, CA (US)

Correspondence Address:
WOMBLE CARLYLE SANDRIDGE & RICE, PLLC
ATTN: PATENT DOCKETING 32ND FLOOR, P.O. BOX 7037
ATLANTA, GA 30357-0037

(73) Assignee: **Immersion Corporation**, San Jose, CA (US)

(21) Appl. No.: **11/748,219**

(22) Filed: **May 14, 2007**

Related U.S. Application Data

(60) Provisional application No. 60/890,690, filed on Feb. 20, 2007.

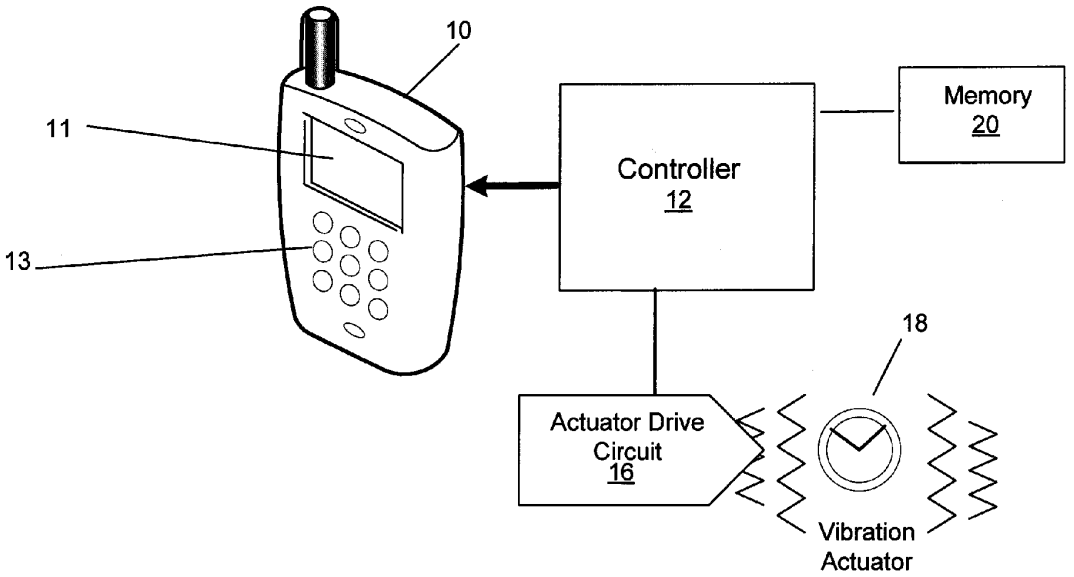
Publication Classification

(51) **Int. Cl.**
G06F 3/041 (2006.01)

(52) **U.S. Cl.** **345/173**

(57) **ABSTRACT**

A haptic feedback system that includes a controller, a memory coupled to the controller, an actuator drive circuit coupled to the controller, and an actuator coupled to the actuator drive circuit. The memory stores at least one haptic effect that is executed by the controller in order to create a haptic effect.



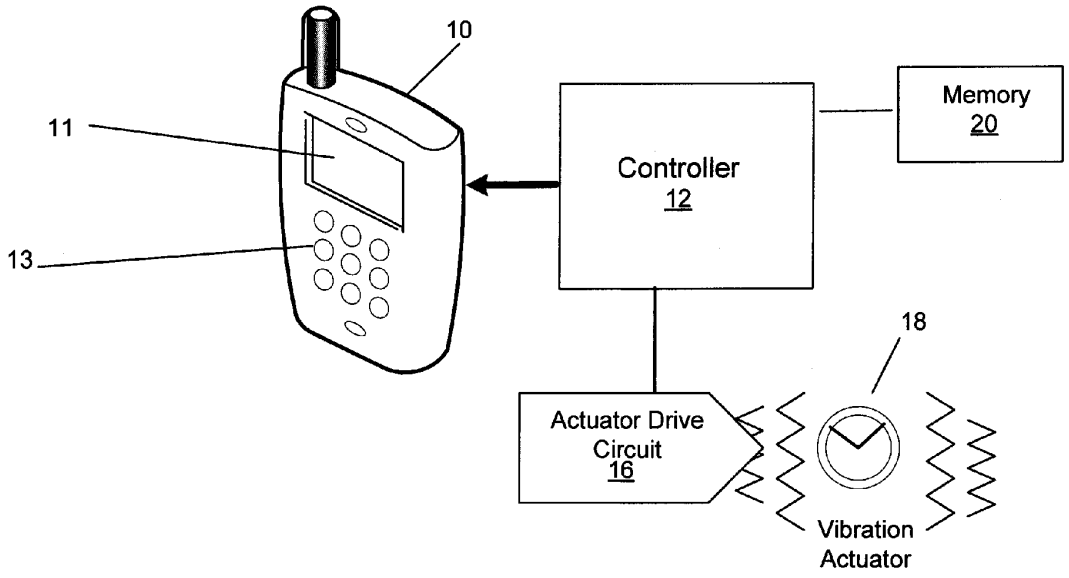

10
11
13
Controller
12
Memory
20
18
Actuator Drive
Circuit
16
Vibration
Actuator

Fig. 1

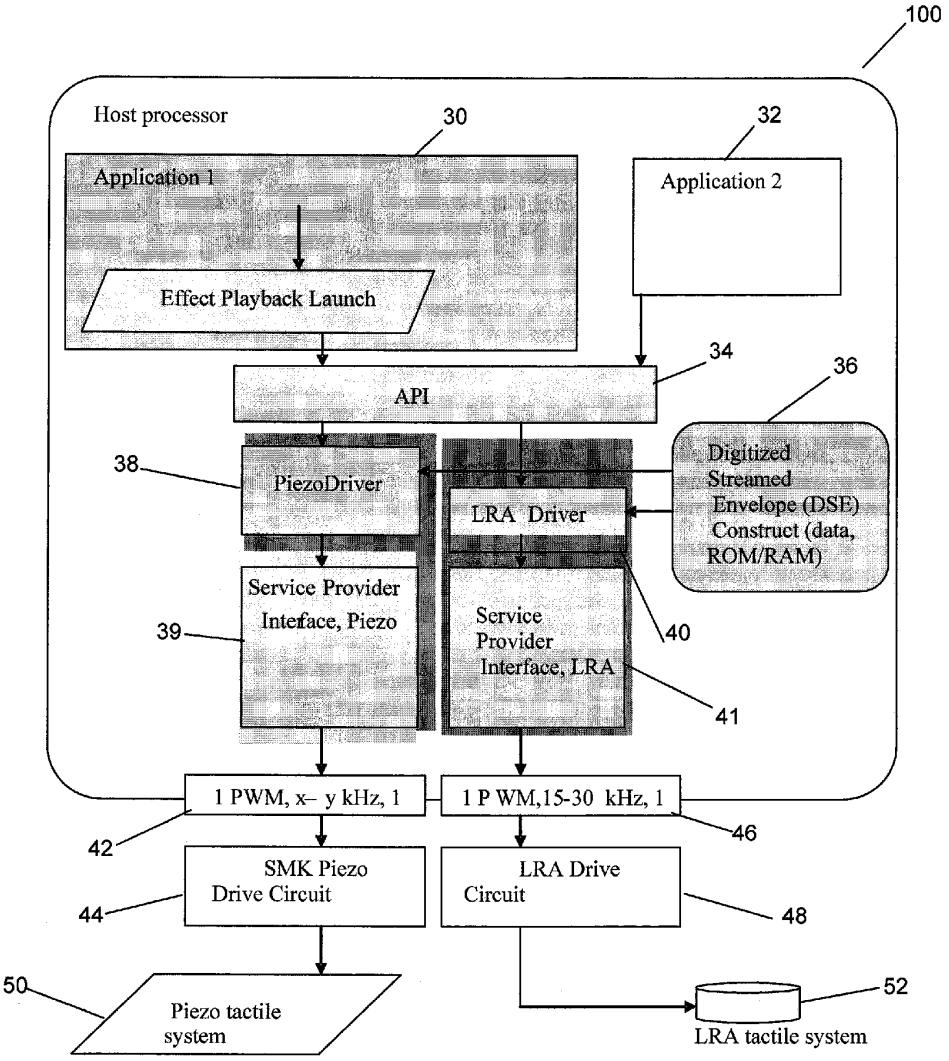

100
Host processor
30
32
Application 1
Application 2
Effect Playback Launch
34
36
API
38
PiezoDriver
LRA Driver
Digitized Streamed Envelope (DSE) Construct (data, ROM/RAM)
Service Provider Interface, Piezo
39
Service Provider Interface, LRA
40
41
1 PWM, x– y kHz, 1
1 P WM,15-30 kHz, 1
42
46
SMK Piezo Drive Circuit
LRA Drive Circuit
44
48
50
Piezo tactile system
52
LRA tactile system

Fig. 2

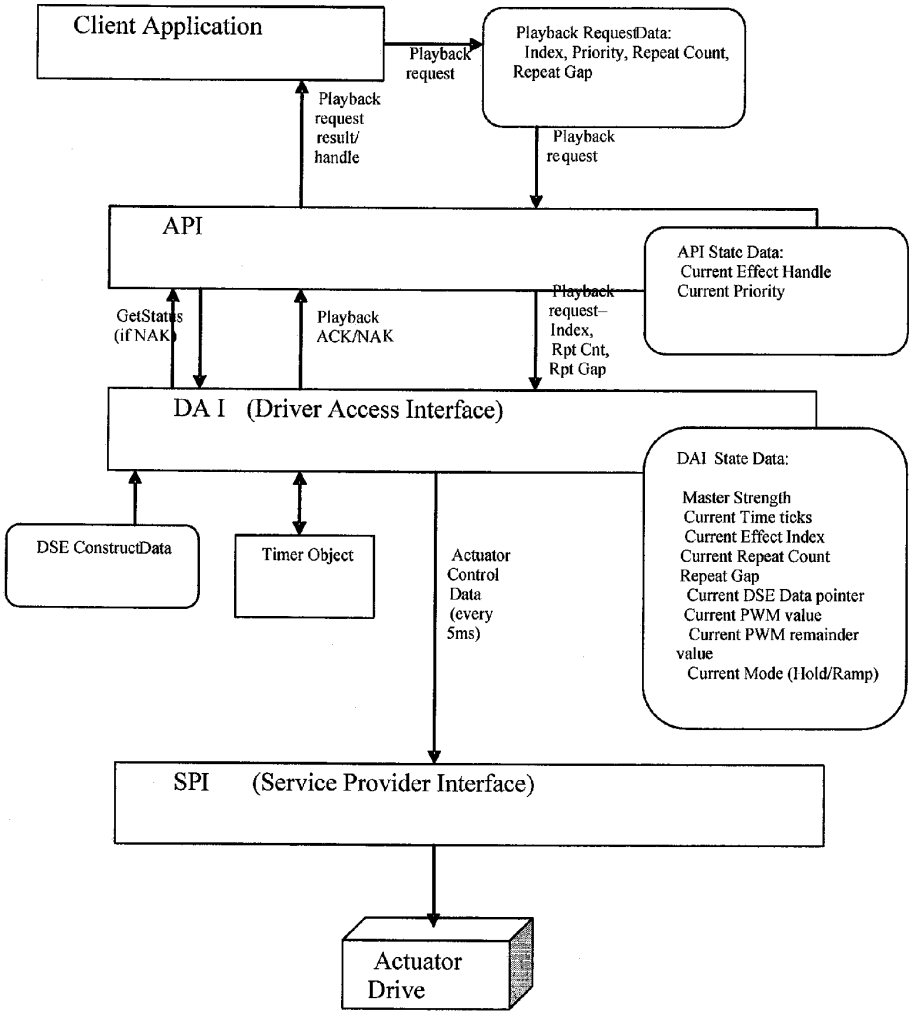

Client Application
Playback request
Playback RequestData:
Index, Priority, Repeat Count,
Repeat Gap
Playback request result/ handle
Playback request
API
API State Data:
Current Effect Handle
Current Priority
GetStatus (if NAK)
Playback ACK/NAK
Playback request– Index, Rpt Cnt, Rpt Gap
DA I (Driver Access Interface)
DAI State Data:
Master Strength
Current Time ticks
Current Effect Index
Current Repeat Count
Repeat Gap
Current DSE Data pointer
Current PWM value
Current PWM remainder value
Current Mode (Hold/Ramp)
DSE ConstructData
Timer Object
Actuator Control Data (every 5ms)
SPI (Service Provider Interface)
Actuator Drive

Fig. 3

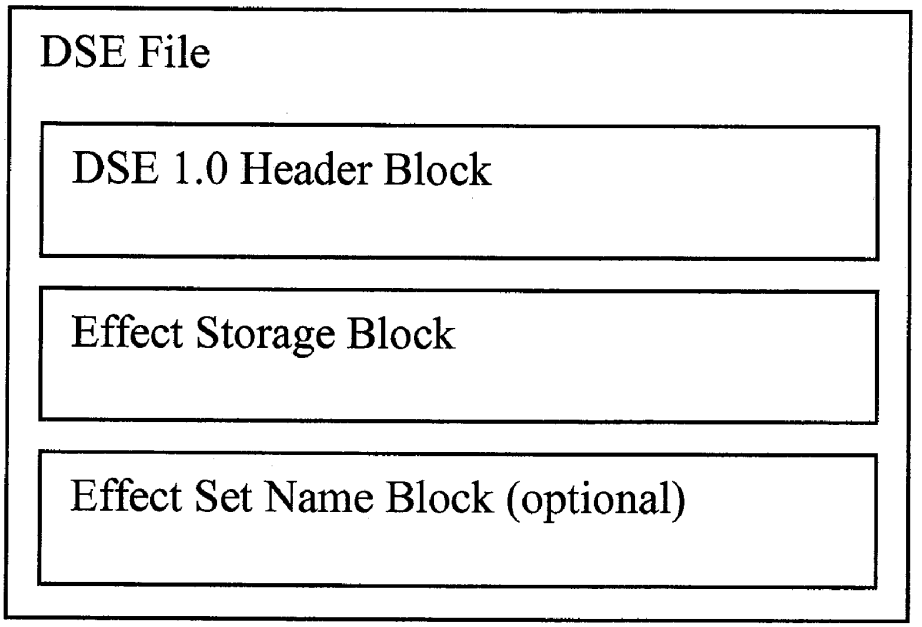

DSE File
DSE 1.0 Header Block
Effect Storage Block
Effect Set Name Block (optional)

Fig. 4

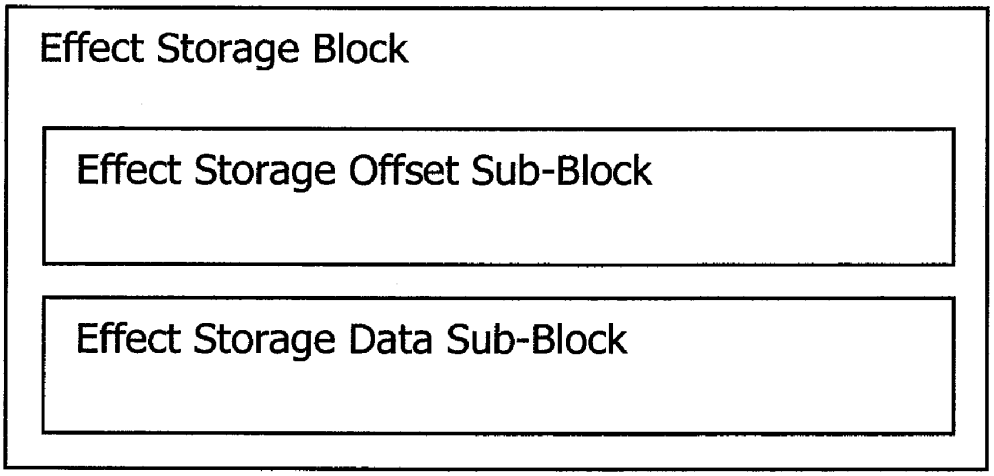

Effect Storage Block
Effect Storage Offset Sub-Block
Effect Storage Data Sub-Block

Fig. 5

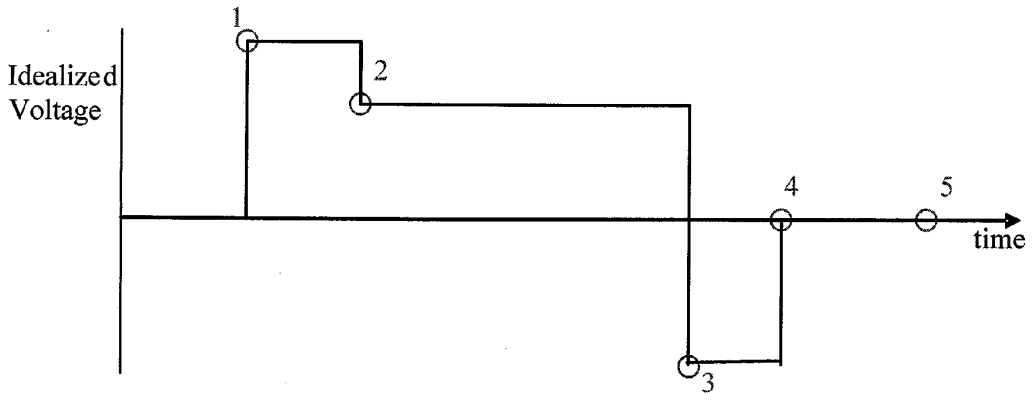


Fig. 6

HAPTIC FEEDBACK SYSTEM WITH STORED EFFECTS

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 60/890,690, filed Feb. 20, 2007.

FIELD OF THE INVENTION

[0002] One embodiment of the present invention is directed to a handheld device with a touchscreen. More particularly, one embodiment of the present invention is directed to a handheld device with a touchscreen that includes a haptic feedback system.

BACKGROUND INFORMATION

[0003] Electronic device manufacturers strive to produce a rich interface for users. Conventional devices use visual and auditory cues to provide feedback to a user. In some interface devices, kinesthetic feedback (such as active and resistive force feedback) and/or tactile feedback (such as vibration, texture, and heat) is also provided to the user, more generally known collectively as "haptic feedback." Haptic feedback can provide cues that enhance and simplify the user interface. Specifically, vibration effects, or vibrotactile haptic effects, may be useful in providing cues to users of electronic devices to alert the user to specific events, or provide realistic feedback to create greater sensory immersion within a simulated or virtual environment.

[0004] Haptic feedback has also been increasingly incorporated in portable electronic devices, such as cellular telephones, personal digital assistants (PDAs), portable gaming devices, and a variety of other portable electronic devices. For example, some portable gaming applications are capable of vibrating in a manner similar to control devices (e.g., joysticks, etc.) used with larger-scale gaming systems that are configured to provide haptic feedback. Additionally, devices such as cellular telephones and PDAs are capable of providing various alerts to users by way of vibrations. For example, a cellular telephone can alert a user to an incoming telephone call by vibrating. Similarly, a PDA can alert a user to a scheduled calendar item or provide a user with a reminder for a "to do" list item or calendar appointment.

[0005] Increasingly, portable devices are moving away from physical buttons in favor of touchscreen only interfaces. This shift allows increased flexibility, reduced parts count, and reduced dependence on failure-prone mechanical buttons and is in line with emerging trends in product design. When using the touchscreen input device, a mechanical confirmation on button press or other user interface action can be simulated with haptics.

[0006] Some known devices modify or generate haptic effects in real-time or "on the fly". Although this allows a wide variety of haptic effects to be generated, it may require a substantial amount of processing power and may not facilitate rapid development of new devices because the wide variety of possible haptic effects must be compatible with the hardware interface of the device.

[0007] Based on the foregoing, there is a need for an improved system and method for generating haptic effects for a device.

SUMMARY OF THE INVENTION

[0008] One embodiment is a haptic feedback system that includes a controller, a memory coupled to the controller, an actuator drive circuit coupled to the controller, and an actuator coupled to the actuator drive circuit. The memory stores at least one haptic effect that is executed by the controller in order to create a haptic effect. The stored haptic effect, unlike real-time generated haptic effects, reduces the required processing power.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. **1** is a block diagram of a cellular telephone in accordance with one embodiment.

[0010] FIG. **2** is a block diagram of the system architecture of a haptic feedback system in accordance with one embodiment.

[0011] FIG. **3** is a block diagram of the software architecture in accordance with one embodiment.

[0012] FIG. **4** is a block diagram of the file format used in one embodiment for a digitized steam envelope construct.

[0013] FIG. **5** is a block diagram of an Effect Storage Block in accordance with one embodiment.

DETAILED DESCRIPTION

[0014] FIG. **1** is a block diagram of a cellular telephone **10** in accordance with one embodiment. Telephone **10** includes a screen **11** and keys **13**. In one embodiment, keys **13** are mechanical type keys. In another embodiment, keys **13** can be implemented by a touchscreen or other type of touch sensitive surface so that keys **13** are touchscreen keys, or can be implemented using any method. Internal to telephone **10** is a haptic feedback system that generates vibrations on telephone **10**. In one embodiment, the vibrations are generated on the entire telephone **10**. In other embodiments, specific portions of telephone **10** can be haptically enabled by the haptic feedback system, including individual keys of keys **13**, whether the keys are mechanically oriented, touchscreen, or some other type of implementation.

[0015] The haptic feedback system includes a controller **12**. Coupled to controller **12** is a memory **20** and an actuator drive circuit **16**, which is coupled to a vibration actuator **18**. Although the embodiment of FIG. **1** is a cellular telephone, embodiments of the present invention can be implemented with any type of handset or mobile/portable device, or any device that uses an actuator to generate vibrations. Further, the haptic feedback system may be implemented as a development board that allows manufacturers of haptically enabled handsets to perform rapid prototypes.

[0016] Controller **12** may be any type of general purpose controller or processor, or could be a controller specifically designed to provide haptic effects, such as an application-specific integrated circuit ("ASIC"). Controller **12** may be the same controller/processor that operates the entire telephone **10**, or may be a separate controller. Controller **12** can decide what haptic effects are to be played and the order in which the effects are played based on high level parameters. In general, the high level parameters that define a particular haptic effect include magnitude, frequency and duration.

[0017] Controller **12** outputs the control signals to drive circuit **16** which includes electronic components and circuitry used to supply actuator **18** with the required electrical current and voltage to cause the desired haptic effects. Actuator **18** is a haptic device that generates a vibration on telephone **10**. Actuator **18** can include one or more force applying mechanisms which are capable of applying a vibrotactile force to a user of telephone **10** (e.g., via the housing of telephone **10**). Actuator **18** may be, for example, an electromagnetic actuator, an Eccentric Rotating Mass ("ERM") in which an eccentric mass is moved by a motor, a Linear Resonant Actuator ("LRA") in which a mass attached to a spring is driven back

and forth, or a "smart material" such as piezoelectric, electro-active polymers or shape memory alloys. Memory **20** can be any type of storage device, such as random access memory ("RAM") or read-only memory ("ROM"). Memory **20** stores instructions executed by controller **12**. Memory **20** may also be located internal to controller **12**, or any combination of internal and external memory.

[0018] FIG. **2** is a block diagram of the system architecture of a haptic feedback system **100** in accordance with one embodiment. System **100** includes two or more applications **30** and **32** that call, via software commands, for a specific stored haptic effect to be played. Application program interface ("API") **34** interprets the software commands and outputs to two parallel paths, one for each type of actuator. In other embodiments, one or more paths may be implemented. In FIG. **2**, the left-side path is for a piezo actuator **50** while the right-side path is for an LRA actuator **52**.

[0019] Coupled to each path is a digitized stream envelope ("DSE") construct **36** which includes stored haptic effects. In one embodiment, four different haptic effects are stored, but any number may be stored. Each path includes a driver **38** and **40** and a service provider interface **39** and **41** that include information on the specific type of actuator for the path. This allows the API to be hardware/actuator independent.

[0020] Each path further includes, for each actuator, a drive signal **42** and **46** and an electrical drive circuit **44** and **48**. Finally, actuators **50** and **52** generate the vibration or other desired haptic effect.

[0021] In one embodiment, the piezo actuator driver expects a differential sine wave control signal. One embodiment generates a sine wave in software at a 5 kHz rate and outputs that as a pulse width modulation ("PWM") signal. A simpler control signal may be used to achieve the same feel on the device. However, in one embodiment, the piezo circuit requires a PWM signal with fixed frequency in the 20-50 kHz range, whose duty cycle is expected to be updated every 200 μs.

[0022] In one embodiment, there are at least 3 possible physical configurations for the LRAs:

[0023] 1. Case mounting—In this configuration, the LRAs are mounted rigidly to the interior of the device casing. The LRA is driven at its resonant frequency (from 175 Hz to 185 Hz), and the entire device is actuated.

[0024] 2. Screen mounting—matched frequency. The screen is floated on a suspension, mechanically isolated from the casing, and the LRAs are rigidly mounted to the screen. The LRAs is typically driven at the LRA resonant frequency, and the suspension is tuned to reinforce that frequency.

[0025] 3. Screen mounting—bi-modal. The screen is floated on a suspension, mechanically isolated from the casing, and the LRAs are rigidly mounted to the screen. The LRA can be driven at the LRA resonant frequency in which case the suspension is tuned to transmit most of the vibration to the casing. The LRA can also be driven at the system's natural frequency, with the suspension tuned to a higher frequency than the LRA's own resonant frequency (typically approx. 500 Hz). In this configuration, the LRA can be used both as an event alerting system (Silent/Manner mode vibration alerter) or as a touchscreen feedback system (for button press tactile confirmation).

[0026] In one embodiment, many variants of the known VibeTonz® LRA drive circuit from Immersion Corp. can be used. Generally, these drive circuits require one PWM at 20-50 kHz fixed frequency, variable duty cycle, 8-bit duty cycle resolution, 50%=no output, with a fine granularity in the 20-30 kHz range (as the carrier frequency is a multiple of the LRA drive frequency in many designs). The circuit also requires a General Purpose Output ("GPO") for AMP_ENABLE control.

[0027] Alternately, an even simpler amp circuit could be used, because a 5 or 8 kHz output rate from the driver would allow output of sine or square waveforms at the LRA resonant frequency. This eliminates the need to fine-tune the PWM frequency, as the resonant frequency would itself be encoded.

[0028] FIG. **3** is a block diagram of the software architecture in accordance with one embodiment. One embodiment of the architecture includes the following:

[0029] 1. No custom effect playback for applications. Only pre-defined effects ("stored haptic effects"), stored in Digitized Streaming Envelope ("DSE") constructs stored in the driver software can be played back.

[0030] 2. No real-time generation. Effect playback is based on a pre-recorded control signal, not generated in real-time. The control signal is computed using an effect design tool.

[0031] 3. Multiple application support. Multiple applications could register with the API simultaneously. Supporting multiple applications requires some form of API client marshaling, where the API must determine whose request is most important. Two approaches are possible:

[0032] a. Last caller wins. When multiple applications try to use the vibration resource simultaneously, the last caller interrupts whatever was playing before and its effect plays.

[0033] b. Priority scheme. The API could support a concept of high/medium/low priority on effect playback launch. When launching an effect, the caller specifies the priority to be used. Playback succeeds when priority is equal to or higher than the current effect's playback priority.

[0034] 4. Designed for portability. Like VibeTonz®, ANSI C only, no dynamic memory allocation.

[0035] In one embodiment, the Driver Access Interface ("DAI") consists of lower-level functions that provide the bulk of the API functionality. In one embodiment, the DAI is designed to be implemented as a functional interface, which is easily wrapped in a serial protocol. The driver is a timed loop that executes the following commands:

[0036] 1. Look at API/Driver shared memory. If playback is scheduled, retrieve current DSE pointer from memory, along with any other required driver/effect state information.

[0037] 2. If state is "playing effect", decode DSE, extract PWM value to be applied to PWM. Then write to PWM. Sleep until next sample.

[0038] 3. If state is "finished iteration check for repeat", and if effect is to be repeated, decrement repeat value, set up gap timing and sleep until gap time expires.

[0039] In one embodiment, DSE construct **36** of FIG. **2** is a set of magnitudes, or strengths, over time, for a number of effects. The samples are stored in a lossless compact format. Files containing DSE information use the .dse file extension.

[0040] FIG. **4** is a block diagram of the file format used in one embodiment for the DSE construct. The DSE 1.0 Header Block contains file format version information, number of effects, location of the Effect Storage Block, location of the Effect Set Name Block, and file size. The Effect Storage Block contains all the effect definitions stored in the DSE file. The Effect Set Name Block, which is optional, contains the name of the effect set.

[0041] FIG. **5** is a block diagram of the Effect Storage Block, in accordance with one embodiment, which has two sub-blocks: an Effect Storage Offset Sub-Block and an Effect Storage Data Sub-Block.

[0042] The Effect Storage Offset Sub-Block is an array of offsets, one offset for each effect. Each offset occupies two bytes with the least-significant byte at the lower memory address. The number of effects is stored in the DSE file header. The offset, in bytes, specifies where in the Effect Storage Data Sub-Block the effect's definition begins relative to the start of the Effect Storage Data Sub-Block. The size of the Effect Storage Offset Sub-Block is 2*EFFECTCOUNT. The EFFECTCOUNT comes from the DSE file header. The Effect Storage Data Sub-Block stores the effect definitions.

[0043] In one embodiment, the DSE contains magnitude control information over time. An idealized voltage, from −127 to 127, is modulated over time to drive a vibration actuator. For AC control (e.g., LRAs), other parts of the system (driver software, electronics) are responsible for synthesizing the AC signal, and the DSE represents the maximum value attained, per cycle, by the AC control signal.

[0044] FIG. **6** illustrates a typical short control signal or stored haptic effect where the critical data to be encoded is clearly identified in accordance with one embodiment. Unlike known systems that generate haptic effects in real-time by generating haptic effects from a plurality of high level parameters, the haptic effect in FIG. **6** is predefined with the low level haptic parameters such as voltage levels and time duration.

[0045] Point **1** is the start of the haptic effect, which is always assumed to start at time t=0 ms. Assume that idealized voltage level is 127. The time between point **1** and point **2** is the kickstart pulse. Point **2**, occurring at, for example, time=20 ms, identifies the start of the sustain period, with voltage level **80** applied. This level is applied until point **3** at time=100 ms, for example, where a braking pulse starts, with voltage at −127. At point **4**, time t=120 ms, voltage drops to 0. An additional zero point, point **5**, is placed at t=150 ms to encode a certain amount of silence—useful when the effect is designed to be repeated (the same effect can be obtained by specifying a gap value in a DAI/API call). Table 1 below is an example of how the critical points and timings are encoded in one embodiment.

TABLE 1

Byte #	Data	Meaning
0	127	Apply voltage level 127 . . .
1	20	For 20 ms.
2	80	Apply voltage level 80 . . .
3	80	For 80 ms.
4	−127	Apply voltage level −127 . . .
5	20	For 20 ms.
6	0	Apply voltage level 0 . . .
7	30	For 30 ms.
8	0	Apply voltage level 0 . . .
9	0	For 0 ms - this means, "end of digitized streamed effect definition".

[0046] In one embodiment a slightly more complex encoding that incorporates a slope encoding is shown in Tables 2 and 3 below. Regarding table 2, the following applies:

[0047] 1. Data is organized in voltage/time pairs.

[0048] 2. Time is relative, not absolute. "Apply X voltage for the next Y milliseconds", and not "Apply voltage X starting at time=Y".

TABLE 2

Generalized Voltage/Time Pair Encoding, "Set & Hold" Pair:

Byte #	Bits	Data	Meaning
0	7	Ramp/nHold	0 => Hold this voltage level constant for this duration
0	6 . . . 0	Voltage/2	Voltage level to be applied, divided by 2. Use a single left shift command and cast to a signed 8-bit integer to obtain the desired voltage level between −127 and 127. The driver should always add 1/−1 to the result unless the value is 0.
1	7 . . . 0	Time, ms/5	Time, in 5 ms increments. For best results, the driver code should run the control loop at 200 Hz. Maximum time that can be encoded = 255 × 5 ms = 1.275 sec. For longer durations, create a sequence of voltage/time pairs.

TABLE 3

Generalized Voltage/Time Pair Encoding, "Set & Ramp" Pair:

Byte #	Bits	Data	Meaning
0	7	Ramp/nHold	1 => Start at the voltage level and ramp according to the enclosed parameters for the specified duration
0	6 . . . 0	Voltage/2	Voltage level to be applied at time 0, divided by 2. Use a single left shift command and cast to a signed 8-bit integer to obtain the desired voltage level between −127 and 127. The driver should always add 1/−1 to the result unless the value is 0.
1	7 . . . 0	Time, ms/5	Time, in 5 ms increments. For best results, the driver code should run the control loop at 200 Hz. Maximum time that can be encoded = 255 × 5 ms = 1.275 sec. For longer durations, create a sequence of voltage/time pairs.

TABLE 3-continued

Generalized Voltage/Time Pair Encoding, "Set & Ramp" Pair:

Byte #	Bits	Data	Meaning
2	7 . . . 0	Slope	−128, or −127 to 127. Whole part of the slope value, representing whole PWM steps to be taken every 5 ms. −128 represents a Slope of "negative 0", versus "positive 0", which is 0. More on slope below.
3	7 . . . 0	SlopeFrac	0 . . . 255. Fractional part of the slope value, expressed as a fraction of 256. More on slope below.

[0049] Using Slope and SlopeFrac

The driver can use these values in the following way:

[0050] pwm=Voltage<<1; /*Initial PWM value*/

[0051] pwm_rem=0; /*Initial PWM remainder*/

In 5 ms loop:

```
If( (pwm__rem + SlopeFrac) < pwm__rem ) /* this remainder is about to
rollover */
{
Slope >= 0 ? pwm++ : pwm--;
}
pwm__rem += SlopeFrac;
if( −128 != Slope) pwm += Slope;
```

[0052] Effect Name Block, which is optional, has one sub-block: an Effect Name Data Sub-Block.

[0053] Several embodiments are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations of are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

What is claimed is:

1. A haptic feedback system comprising:

a controller;

a memory coupled to said controller;

an actuator drive circuit coupled to said controller; and

an actuator coupled to said actuator drive circuit;

wherein said memory stores at least one haptic effect.

2. The haptic feedback system of claim **1**, wherein said haptic effect comprises a plurality of discrete magnitude points.

3. The haptic feedback system of claim **2**, wherein a magnitude is held constant over a time period.

4. The haptic feedback system of claim **2**, wherein a magnitude is modulated over a time period as a function of time.

5. The haptic feedback system of claim **2**, wherein said discrete magnitude points comprise a start point, a sustain point, a brake point, and a zero point.

6. The haptic feedback system of claim **1**, wherein said haptic effect is a pre-computed haptic effect.

7. The haptic feedback system of claim **6**, wherein said haptic effect is generated with an effect design tool.

8. The haptic feedback system of claim **1**, wherein said actuator is a Linear Resonant Actuator.

9. The haptic feedback system of claim **1**, wherein said actuator is a piezo actuator.

10. The haptic feedback system of claim **1**, wherein the haptic effect is stored as a digitized streamed envelope construct.

11. A method of generating a haptic effect comprising:

retrieving a stored haptic effect;

generating drive signals based on the stored haptic effect; and

applying the drive signals to an actuator.

12. The method of claim **11**, wherein the haptic effect comprises a plurality of discrete magnitude points.

13. The method of claim **12**, wherein a magnitude is held constant over a time period.

14. The method of claim **12**, wherein a magnitude is modulated over a time period as a function of time.

15. The method of claim **12**, wherein the discrete magnitude points comprise a start point, a sustain point, a brake point, and a zero point.

16. The method of claim **11**, wherein the haptic effect is a pre-computed haptic effect.

17. The method of claim **11**, wherein the haptic effect is generated with an effect design tool.

18. The method of claim **11**, wherein the actuator is a Linear Resonant Actuator.

19. The method of claim **11**, wherein the actuator is a piezo actuator.

20. The method of claim **11**, wherein the haptic effect is stored as a digitized streamed envelope construct.

21. A computer readable medium having instructions stored thereon that, when executed by a processor, cause the processor to:

retrieve a stored haptic effect;

generate drive signals based on the stored haptic effect; and

apply the drive signals to an actuator.

22. The computer readable medium of claim **21**, wherein the haptic effect comprises a plurality of discrete magnitude points.

23. The computer readable medium of claim **22**, wherein a magnitude is held constant over a time period.

24. The computer readable medium of claim **22**, wherein a magnitude is modulated over a time period as a function of time.

25. The computer readable medium of claim **22**, wherein the discrete magnitude points comprise a start point, a sustain point, a brake point, and a zero point.

26. The computer readable medium of claim **21**, wherein the haptic effect is a pre-computed haptic effect.

27. The computer readable medium of claim **22**, wherein the haptic effect is generated with an effect design tool.

28. The computer readable medium of claim **22**, wherein the actuator is a Linear Resonant Actuator.

29. The computer readable medium of claim **22**, wherein the actuator is a piezo actuator.

30. The computer readable medium of claim **22**, wherein the haptic effect is stored as a digitized streamed envelope construct.

* * * * *