(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 August 2016 (11.08.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/123768 A1

(51) International Patent Classification:
G06F 9/445 (2006.01)

(21) International Application Number:
PCT/CN2015/072263

(22) International Filing Date:
4 February 2015 (04.02.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **GLOBAL SILICON LIMITED** [CN/CN]; 905 Silvercord Tower 2, 30 Canton Road, Tsimshatsui, Kowloon, Hong Kong (CN).

(72) Inventor: **COLMER, Morgan James**; 905 Silvercord Tower 2, 30 Canton Road, Tsimshatsui, Kowloon, Hong Kong (CN).

(74) Agent: **HENSEN INTELLECTUAL PROPERTY FIRM**; 10H, Shangbu Building, No.68 Nanyuan Road, Futian, Shenzhen, Guangdong 518000 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

WO 2016/123768 A1

(54) Title: UPDATING FIRMWARE WITH AUDIO

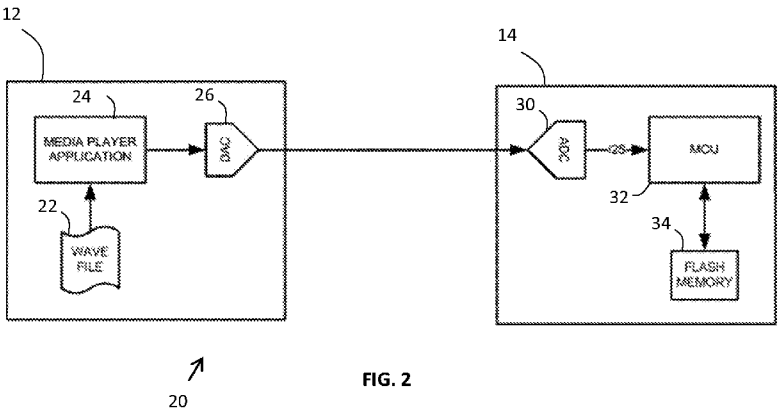


FIG. 2

(57) Abstract: A method for modifying a system configuration on a programmable device is disclosed comprising generating a modify file in a digital media content format; storing the modify file on a digital media content player; connecting the digital media content player to the programmable device; playing the modify file with the digital media content player for transmitting data for modifying the system configuration of the programmable device.

Description

Title of Invention: UPDATING FIRMWARE WITH AUDIO

Technical Field

[1] The invention relates generally to a method and apparatus for updating firmware in electronic consumer devices such as programmable input/output (I/O) device peripherals, e.g. audio/video devices, and more particularly to a method and apparatus for updating firmware in such electronic devices via digital audio files.

Background Art

[2] Consumer demand has increased for programmable input/output (I/O) peripherals such as audio/video devices to play digital media content stored on computing platforms such as smart/feature phones, tablet computers, laptop computers, desktop computers, standalone media players, smart televisions (TV), portable media players, and the like. Such programmable I/O peripherals typically have firmware programmed into persistent memory that may require updating or upgrading with new versions of the firmware to incorporate a new feature, functionality, compatibility, modification, enhancements, or the like.

[3] In view of the number of different computing platforms, operating systems and new media, manufacturers and developers typically spend substantial time and expense to develop proprietary applications for each type of computing platform and operating systems being used to deliver firmware updates or upgrades.

[4] US2010/0161689 discloses method for updating/modifying/upgrading the system configuration of a stand-alone non-network connectible device. The method is performed while a portable device is docked to the non-network connectible device. System configuration update information is embedded within the ordinary data content that is received from the portable device during the routine operation of the stand-alone non-network connectible device. Upon receiving the embedded system configuration update information the non-network connectible device decrypts the embedded information and updates the system configuration accordingly. Generation of the embedded system configuration update information is performed in the portable docking device, and installation is performed on the non-network connectible device without requiring a network connection in the stand-alone device.

[5] The method taught by US2010/0161689 suffers from problems associated with low frequency offsets. Furthermore, if there is a phase inversion in the data, the data encoding will fail, and it cannot use compressed audio formats.

[6] There is therefore a need to address or at least alleviate or mitigate some of the above issues to reduce the overhead associated with updating or upgrading firmware.

Disclosure of Invention

Solution to Problem

Technical Solution

[7] In a first main aspect, the invention provides a method for modifying a system configuration of a programmable device comprising: providing a modify file in a digital media content format to an electronic device adapted to play a digital media content file; connecting said electronic device to the programmable device; and playing the modify file with the electronic device for transmitting data in an analog signal format to said programmable device for modifying the system configuration of the programmable device.

[8] In a second main aspect, the invention provides an electronic system for modifying a system configuration of a programmable device, said electronic system being capable of playing a digital media content file, the electronic system comprising: memory for storing a modify file in a digital media content format; means for connecting to a programmable device; and means for playing the modify file and formatting data of said modify file to an analog signal format to transmit said data in said analog signal format to said programmable device.

[9] In a third main aspect, the invention provides a method of formatting a digital media content file into an audio signal format comprising: converting data of said media content file from an m bit code space to respective n bit channel symbols, where m and n are positive integers, $m < n$ and the channel symbols are chosen from an available set of channel symbols to have zero DC content.

[10] In a fourth main aspect, the invention provides a computer readable medium comprising program code executable by a processor of an electronic device to implement the steps of the methods of the first or third main aspects of the invention.

Brief Description of Drawings

Description of Drawings

[11] The accompanying drawings incorporated herein and forming a part of the specification illustrate several aspects of the present invention and, together with the description, serve to explain the principles of the invention. While the invention will be described in connection with certain embodiments, there is no intent to limit the invention to those embodiments described. On the contrary, the intent is to cover all alternatives, modifications and equivalents as included within the scope of the invention as defined by the appended claims. In the drawings:

[12] FIG. 1 is a simplified schematic view of a digital media content player connected to a programmable device in accordance with an embodiment of the invention;

[13] FIG. 2 is a simplified block diagram of the basic signal path from the digital media

content player to the programmable device in accordance with an embodiment of the invention;

[14] FIG. 3 is a flow chart in accordance with an embodiment of the invention;

[15] FIG. 4 is a graph showing a simulation of a received sampled signal in accordance with an embodiment of the invention;

[16] FIG. 5 is a graph showing an ADC digital filter frequency response in accordance with an embodiment of the invention;

[17] FIG. 6 is a simplified block diagram of the soft-decision threshold with physical layer calibration phase in accordance with an embodiment of the invention;

[18] FIG. 7 is a graph showing the upper and lower thresholds assigned during the calibration process in accordance with an embodiment of the invention;

[19] FIG. 8 is a truth table for the differential encoder and decoder in accordance with an embodiment of the invention;

[20] FIG. 9 is a diagram showing two possible input signals to the differential decoder that will result in the same decoded channel symbol in accordance with an embodiment of the invention;

[21] FIG. 10 is a simplified block diagram of the differential encoder in accordance with an embodiment of the invention;

[22] FIG. 11 is a simplified block diagram of the differential decoder in accordance with an embodiment of the invention;

[23] FIG. 12 is a simplified block diagram of the four-to-six demodulator and four-to-six modulator in accordance with an embodiment of the invention;

[24] FIG. 13 is a four-to-six encoding/decoding table in accordance with an embodiment of the invention;

[25] FIG. 14 is a graph showing a four-to-six modulation example in accordance with an embodiment of the invention;

[26] FIG. 15 is a graph showing the sync-symbol at the beginning of each 64 symbol packet in accordance with an embodiment of the invention;

[27] FIG. 16 is a channel code and channel symbol table in accordance with an embodiment of the invention;

[28] FIG. 17 is diagram showing the general data packet structure in accordance with an embodiment of the invention;

[29] FIG. 18 is timing diagram showing the physical layer calibration of sync symbol in accordance with an embodiment of the invention;

[30] FIG. 19 is a diagram showing the sequence in accordance with an embodiment of the invention;

[31] FIG. 20 is a diagram showing the general packet format structure in accordance with an embodiment of the invention;

[32] FIG. 21 is a diagram showing the packet data format structure at the receiver in accordance with an embodiment of the invention;

[33] FIG. 22 is 4-bit data fields table in accordance with an embodiment of the invention;

[34] FIG. 23 is a file control packet table in accordance with an embodiment of the invention;

[35] FIG. 24 is a file CRC packet table in accordance with an embodiment of the invention;

[36] FIG. 25 is a left and right audio channel data table in accordance with an embodiment of the invention;

[37] FIG. 26 is a graph of a simulation of a perfect recovery of 2 symbols from the analog data in accordance with an embodiment of the invention;

[38] FIG. 27 is a graph of a white data undetermined sample in accordance with an embodiment of the invention;

[39] FIG. 28 is a graph of results of sample discriminator in accordance with an embodiment of the invention; and

[40] FIG. 29 is a simplified block diagram of forward error correction in accordance with an embodiment of the invention.

Mode for the Invention

Mode for Invention

[41] A protocol and system is disclosed to enable transmission of firmware upgrades or the like to consumer electronic devices from electronic devices such as computing platforms or smart/feature phones without the need to develop proprietary applications for each type of platform being used to deliver the firmware upgrade.

[42] The system uses custom encoded, uncompressed wave files (.WAV) to pass data in a specially modulated signal format to the device being upgraded using an analog audio signal over a connection between the computing platform and the programmable consumer electronic device. The connection may be a wireless connection, but, in preferred embodiments, the connection comprises an audio cable connection and, more particularly, a stereo cable connected from the smart phone, for example, to the device being upgraded. In an embodiment, the consumer electronic device is a speaker box or system which is also equipped with a microcontroller unit (MCU) and a 3.5mm stereo input jack. Most computing platforms capable of playing a digital media content file such as a .WAV file have a compatible output connector such as , for example, a headphone connector.

[43] Referring to FIG. 1, a simplified schematic view 10 is shown of an electronic device such as a digital media content player 12 connected to a programmable device 14 in accordance with an embodiment of the invention. The player 12 may be connected to

the programmable device 14 via wire or wireless means such as by stereo analog audio cable, Bluetooth™, or the like.

[44] Electronic devices or computing platforms which may be used to deliver the wave file include, are not limited to, smart/feature phones, tablet computers, laptop computers, desktop computers, standalone media players, smart TVs, MP3 players, etc. The advantage to the consumer is that all 'smart' media player type products are compatible with this system, because the system requires no special software application or hardware and is operating system (OS) agnostic. The advantage to the manufacturer or developer is that the development overheads are minimised. Many modern speaker systems have MCUs included that have some limited digital signal processing (DSP) capabilities and have access to the digitized audio stereo input signal stream. These speakers typically also have an auxiliary analog input port, often called a line-in, to allow wired audio sources to be accommodated.

[45] Within the speaker, the line in analog signal is digitized with an analog to digital converter (ADC). Within most media playback devices such as handsets, there is the reverse process occurring, the digital audio that is rendered by the media player software application is turned into an analog signal and fed to an analog audio output, often comprising a headphone socket.

[46] The system description given here is for a specific embodiment, however, it will be appreciated that the basic principles apply even if the modulation scheme, data rates, frame structure or purpose of the data exchange are modified to suit other applications. In practice, the user would: download a short wave file with the special encoding included; place the target system into a special firmware upgrade mode, although this is not an essential step; and then just play the file in the native media player application on the electronic device. In doing so, the data of the modify wave file is transmitted to the programmable device to thereby upgrade its firmware. It will also be appreciated that this feature may be automatic. After the file has been played, the target device's firmware is upgraded using the transmitted data.

[47] FIG. 2 is a simplified block diagram 20 of the basic signal path from the digital media content player to the programmable device in accordance with an embodiment of the invention. The digital media content player 12 comprises a wave file 22, media player application 24, and a digital to analog converter 26. The programmable device 14 comprises an analog to digital converter (ADC) 30 a microprocessor (MCU) 32, and a flash memory 34.

[48] If the handset is playing back a specially encoded data sequence then it is possible to pass digital data to the speaker from playback of a wave file. Because wave files are not compressed and run at their native sample rate, the digital data recovered at the MCU in the speaker will be a close approximation to the original data sequence.

Special encoding techniques are described below that are used to overcome some of the limitations in achieving digital communications, i.e. noise, spectral distortion of the channel and absence of direct clock synchronization.

[49] In this application, the purpose is to allow remote reprogramming of the firmware for the speaker's MCU using the existing small flash memory 34, for example used for storing audio samples, as a temporary buffer storage, the program is located inside the MCU in on-chip flash memory.

[50] FIG. 3 is a flow chart 40 in accordance with an embodiment of the invention. The method in accordance with an embodiment of the invention comprises modifying a system configuration on a programmable device 14 by generating 41 a modify file in a digital media content format. The modify file is then stored 42 on a digital media content player 12, and the digital media content player 12 is connected 44 to the programmable device 14. The digital media content player 12 plays 46 the modify file for transmitting 48 data for modifying the system configuration of the programmable device 14.

[51] FIG. 4 is a graph 50 showing a simulation of a received sampled signal in accordance with an embodiment of the invention. Within the analog domain, digital bits are represented as voltage levels. In the data transmission device there is a media player that will take digital samples in signed 16 bit format and deliver these to the digital to analog converter (DAC) that will reproduce this signal in the analog domain. Within the digital domain, the two possible output values are for 1's: 0x7FFF and for 0's: 0x8000 to give the maximum possible channel SNR. In the analog domain, these may be represented as a number of different values depending upon the device but it is always assumed that the receiver and the transmitter will have matched analog levels. At the receiving device's ADC, these will be once again mapped into digital values but with the bandwidth and noise of the channel imposed. FIG. 4 is a simulation of a typical received sampled signal with a decision threshold of +/- 0.5 applied.

[52] FIG. 5 is a graph 60 showing an ADC digital filter frequency response in accordance with an embodiment of the invention. FIG. 5 shows an ADC digital filter frequency response, in the simulations shown, the spectral distortion has been exaggerated in order to demonstrate that the proposed system will operate correctly in all system configurations. The output frequency response of the media player's DAC also needs to be considered but in general these are equivalent to the above.

[53] FIG. 6 is a simplified block diagram 70 of the soft-decision threshold with physical layer calibration phase in accordance with an embodiment of the invention. On the face of it, turning an analog signal into a binary bit stream would appear to be a trivial task but in this application there are a number of challenges that come from the diversity of physical links available.

[54] At the beginning of every data session there is a physical layer calibration phase, this process is to normalize the data slicing thresholds and establish a baseline for zero. The analog data slicing needs to have hysteresis to reduce the effect of noise manifesting itself as jitter on the digital signal output.

[55] FIG. 7 is a graph 80 showing the upper and lower thresholds assigned during the calibration process in accordance with an embodiment of the invention. During the calibration process, the upper and lower thresholds are preferably assigned to be symmetrical around the baseline value found for zero and are expected to be within the range of +/-25% to +/-50% of the nominal 1 or 0 as shown in FIG. 7.

[56] FIG. 8 is a truth table 90 for the differential encoder and decoder in accordance with an embodiment of the invention. Channel data is send via NRZ differentially encoded serial data in four-to-six modulation (FSM) format. Each FSM data symbol is differentially encoded so that a '1' represents a data transition and a '-1' (or logical '0') represents no data transition. The truth table for the differential encoder and decoder is dependent upon the previous input X[n-1] and the current input X[n] as shown in FIG. 8.

[57] FIG. 9 is a diagram 110 showing two possible input signals to the differential decoder that will result in the same decoded channel symbol in accordance with an embodiment of the invention. In this way data may be in inverted or normal sense and the decoded data symbols are unchanged. The diagram of FIG. 9 shows two possible input signals to the differential decoder that will result in the same decoded channel symbol. It will be appreciated that the previous state determines the polarity of the decoder output, for this reason, every packet has a fixed polarity start sequence (see sync-symbol) that overrides the differential encoder.

[58] FIG. 10 is a simplified block diagram 110 of the differential encoder in accordance with an embodiment of the invention, and FIG. 11 is a simplified block diagram 120 of the differential decoder in accordance with an embodiment of the invention. The equivalent circuit of the differential encoder is show in FIG. 10 where the channel clock is assumed to be 12 KHz, but this is not an inherent limitation. Other frequencies may be used. This is used at the final stage of the encoder before the wave file is written. The equivalent circuit of the differential decoder is show in FIG. 11, the channel clock is also assumed to be 12 KHz. This is the first stage in the decoding process and directly from the audio data received.

[59] FIG. 12 is a simplified block diagram 130 of the four-to-six demodulator and four-to-six modulator in accordance with an embodiment of the invention. Due to the nature of audio analog input and outputs, care must be taken to remove any low frequency content from the data spectrum. This is achieved by four-to-six modulation (FSM). The FSM encoder/decoder will convert data from 4 bit code space into 6 bit channel

symbols (or visa-versa), as shown in FIG. 12. The channel symbols have been chosen to all have zero DC content (i.e. equal numbers of 1's and 0's per symbol). From the 64 symbols available from 6-bit encoding, only 20 have zero DC content. Of these, a subset of 16 have been chosen as the channel symbols for the data 0x0 to 0xF, the remaining 4 symbols can be allocated as optional sub-codes for potential future system developments or expansion. Whilst four-to-six modulation/demodulation is preferred, other size bit code space to different sized bit channel symbols may be utilized. The use of bit code space to bit channel symbol modulation mitigates the effect of any DC blocking circuits of both the digital content media player and the device being upgraded which might otherwise introduce an offset leading eventually to the discrimination thresholds of one or both devices being exceeded.

[60] FIG. 13 is a four-to-six encoding/decoding table 140 in accordance with an embodiment of the invention. Encoding and decoding are exactly juxtaposed so that encoding data to FSM and then decoding back to data will result in the same data value. It is anticipated that this will be achieved with a basic look-up table. The fact that all channel symbols have been chosen to have exactly zero DC content removes the need for any protocol overhead to include dynamic 'merging bits' to push the DC out of band.

[61] FIG. 14 is a graph 150 showing a four-to-six modulation example in accordance with an embodiment of the invention. As an example of the channel coding, FIG. 14 is the example of two pieces of 4-bit data (0x0 and 0x1) being encoded to FSM (analog + digital representation with assumed channel filter properties simulated) and then differentially encoded. In this case the FSM encoded data is 0x04 and 0x09 which becomes 000111 and 001110 after differential encoding. It will be appreciated that data format is most significant bits first.

[62] FIG. 15 is a graph 160 showing the sync-symbol at the beginning of each 64 symbol packet in accordance with an embodiment of the invention. At the beginning of each 64 symbol packet is a unique pair of channel symbols called the sync-symbol. This is used to identify the beginning of each packet and consists of 6 + 6 channel bits set 6 low followed by 6 high as shown in FIG. 15. The sync-symbol may not be differentially encoded as its phase is fixed, during the decoding process, the differential decoder will accept this symbol as the initializing state for the decoder for the whole data packet. The sync-symbol detection process will occur before differential decoding.

[63] FIG. 16 is a channel code and channel symbol table 170 in accordance with an embodiment of the invention. There are 4 additional codes available in the FSM coding which meet the criteria of having zero net DC content. These have been reserved for future use as semaphores or sub-codes decoded over multiple packets or frames. They

are defined as 'not a number' (NAN). In future enhancements to the system, these could be allocated as bit weightings and thus result in sub-code data over successive packets or frames.

[64] FIG. 17 is diagram 180 showing the general data packet structure in accordance with an embodiment of the invention. A data session is defined as a set of packets that form a complete transmission/reception session. Both audio channels have the same PLC and FPC but the GDP packets and CRCs are different to allow the bandwidth of the system to be doubled. The overall file structure per session on each audio channel with 'n' general data packets has the following overall structure, as shown in FIG. 17.

[65] FIG. 18 is timing diagram 190 showing the physical layer calibration of sync symbol in accordance with an embodiment of the invention. The calibration process is the first event that occurs in the data transmission process. The purpose of this delay is to ensure that the output and input stages of the transmitter and receiver have settled down to their zero DC level before the system begins sending data. During this time, the transmitter will send digital silence and the receiver will calibrate the DC level representing zero and establish soft thresholds for states of 1 and 0 during the 5 sync-symbols that are interspersed in the PLC. In this embodiment, the calibration time will be approximately 0.16 seconds long in total divided into 5 sections with 1536 samples (32ms) of digital silence on both audio channels starting with a SYNC symbol also on both audio channels.

[66] When all 5 cycles are added, the sequence as shown in FIG 19 is formed. FIG. 19 is a diagram 200 showing the sequence in accordance with an embodiment of the invention.

[67] FIG. 20 is a diagram 210 showing the general packet format structure in accordance with an embodiment of the invention. Each data packet is atomic and has a fixed format regardless of the type of packet or data usage. There is not a fixed number of packets per file, the FCP (file control packet) contains the file length and is described later. The general packet format is as follows and all packet types have the same structure. After the SYNC symbol, there are 64 4-bit data elements in each packet, after which the sequence continues from the SYNC symbol again until all data packets have been sent. In each packet there are therefore 66 x 6 channel bits = 396 channel bits. Each channel bit is quarter-multiple of the fundamental sampling rate of 48 KHz so each packet takes 33ms to transmit and contains (after decoding) 32 bytes (16 words) of data.

[68] At the receiver the 4-bit data from each packet is then re-ordered into 16-bit data structures as shown in FIG. 21. FIG. 21 is a diagram 220 showing the packet data format structure at the receiver in accordance with an embodiment of the invention. Because there are two audio channels that are active in sending data, the two data

streams need to be later merged into the final file format which is discussed later.

[69] FIG. 22 is 4-bit data fields table 230 in accordance with an embodiment of the invention. The file control packet is the first packet sent after calibration during data transmission and it contains information about the type of file being sent, its size and application. The FCP is transmitted by both audio channels at the same time with the same data, because this packet is not included in the file CRC packet check process, two identical copies must be received and validated before the main data file is received. The packet structure is identical to the general data packet structure but with specific data fields regarding the data file being sent. The FCP also has its own CRC field for validity checking. The 4-bit data fields are defined as follows as shown in FIG. 22, and the file control packet has the layout shown in the table of FIG. 23 of its elements within the packet, data width and bit order in accordance with an embodiment of the invention.

[70] The FCP CRC check is the CRC-16-ANSI standard created by IBM® systems using a check polynomial of 0x8005. This algorithm requires that all data is fed through a polynomial long division process in 16-bit format. This applies to all data in the FCP from D0 to D59 inclusive.

[71] FIG. 24 is a file CRC packet table 250 in accordance with an embodiment of the invention. This is a 32 bit CRC of the entire data manifest with the exception of the CRC packet and the FCP. Each channel (left/right) encodes its own independent 32 bit CRC and each must be separately checked for consistency. The purpose of this data check is reduce the possibility of programming damaged data into the device, potentially preventing further upgrades being possible. The possibility of both channels CRCs being incorrect and the data matching this for both channels is over $4x10^{24}$: 1, and the CRC packet is generated using CRC-32-ANSI, the same open standard using in PNG, PKZIP and many other file types. The check polynomial is 0x04C11DB7 as shown in FIG. 24.

[72] FIG. 25 is a left and right audio channel data table 260 in accordance with an embodiment of the invention. The file format is type specific; in this case the definition is for a firmware binary file used to re-program the MCU applications. The file is 38 Kbytes (38912 bytes) of actual data which is contained within the GDPs. In general the firmware update files will be kept at a constant length regardless of the size of the actual firmware, the full program memory will be accommodated. For the GS123 MCU, there will be 608 general data packets sent to each audio channel (1216 in total) allowing the MCU to be upgraded in under 21 seconds. The general data packets are first assembled upon receipt into 16 x 16-bit blocks and then these are placed though each channel's CRC checking algorithm on a packet-by-packet basis. After passing through the CRC checking algorithm, the packets are stored in external flash memory

in contiguous blocks, following a LEFT channel first, RIGHT channel second ordering; this is the final order of the data for programming the flash memory. Once all 1216 packets have been received, the CRC computation from both audio channels will have arrived at a result, the next packet to arrive on each audio channel is the GDP CRC-32 packet. Once this has been compared with the result computed for each audio channel and found to be correct, the flash burning process may begin and the new code image may be burnt into the MCU's internal flash. Should the CRC be found to be incorrect for either audio channel, all of the data is discarded, the user is notified that the programming has failed and the target system should re-boot.

[73] In view of the fact that the crystal oscillators that generate the clocks on both the handset and the speaker are in no way synchronized, the system must adapt to the variance in clock rates at the receiver. One of the reasons that in this 48 KHz sampled system, the channel clock is a quarter of that is that the oversampling may be used to remove the significance of this cycle slip. The typical accuracy of a crystal oscillator used in a speaker or handset is at least 50ppm accurate, this method will accommodate inaccuracies as much as 100 times that level. If the inaccuracy is as much as 500ppm then this will result is a gained or lost sample occurring at a rate of around 50 times per second. This system may cope with at least 8000 gained or lost cycles per second.

[74] FIG. 26 is a graph 270 of a simulation of a perfect recovery of 2 symbols from the analog data in accordance with an embodiment of the invention. The sample-discriminator is able to cope with +/- one 48 KHz sample error. Essentially any sample transitions that are found to have occurred either one sample early or one sample too late are fixed in the data decoding process and a sample is either inserted or deleted to make the number of received 1's or 0's correct. FIG. 26 shows a simulation of a perfect recovery of 2 symbols from the analog data and the data sequence it produces: 001110 001110.

[75] FIG. 27 is a graph 280 of a white data undetermined sample in accordance with an embodiment of the invention. FIG. 27 is the same data sequence that has been degraded by changing the timing by one sample. This results in some data samples that may not be resolved within the +/-0.5 decision threshold. Because this level of errors falls within the 3-to-5 threshold, the soft-decision unit may correct the data. The white data sample below is an undetermined sample.

[76] FIG. 28 is a graph 290 of results of sample discriminator in accordance with an embodiment of the invention. In this case above, the white sample may be determined by the sample discriminator to be a 0. In the example shown in FIG. 28, the data quality has fallen unacceptably low and the sample discriminator may not always be certain to give the correct result as there are two samples that are either added or removed. The results could be either 001110 001110 or 001110 011110. Upon encountering the

situation or two samples in error, the bit-discriminator must mark the bit as an erasure (0?1110) and later may further process this in the symbol-discriminator.

[77] Due to the encoding of the four-to-six modulation in order to produce an error that will change a valid symbol into another valid symbol, two bits must be changed. In the above example, a symbol has a marked erasure or 0?1110 and this only has two possible states, 011110 and 001110, of which only one is a valid symbol. Upon finding an erasure in a symbol, the symbol discriminator needs to search the valid symbol table for the symbol that matches the other 5 known bits; the erasure may then be replaced with the correct bit. This also gives a basic level of error correction; the system may recover 2 samples in error within 24.

[78] The sample discriminator may fix errors of up to +/-1 sample. The symbol discriminator may fix errors of +/-2 samples. If there are 3 or 4 sample errors then both the sample discriminator and the symbol discriminator will not detect or correct the error but the resulting symbol will be invalid. In the case of +/-3 or 4 samples in error, the error symbol may then be compared to the table of valid FSM symbols with a search algorithm until it arrives at a symbol where on one bit change is needed to produce a correct symbol. Because of the nature of the valid code distribution within the FSM code table, there will only be one valid symbol capable of meeting the criteria of a single bit in error. This increases the error correction capability to 4 samples within 24, or 1 bit in 6.

[79] FIG. 29 is a simplified block diagram 300 of forward error correction in accordance with an embodiment of the invention. Should the data error rate begin to become noticeable then a further enhancement to the system is the forward error correction (FEC). The technique chosen is Reed-Solomon error coding and correction using a Galois Field of GF(2^8) with the following polynomial:

[80] $P_{(x)} = x^8 + x^4 + x^3 + x^2 + 1$

[81] The primitive of this prime polynomial is alpha, defined as 0x02 (00000010). Because they are working with packet sizes of 32 bytes, this lends itself to a 32/28 coding gain, essentially with the loss of 4 bytes of data per packet may allow the correction of up to 1 byte per packet with a very low chance of mis-correction but given that the overall data file has a further very secure CRC then it is possible to correct up to 2 bytes per 32 byte packet.

[82] The file length will become increased by the addition of error correction as there are now 4 party symbols per 28 bytes of data. These parity symbols may be used in a number of ways, they may be used to either detect errors, or they may be used to locate and correct errors but not both at the same time. An example of the limitations is given. If all 4 parity (syndromes) are used for error checking then all 32 bits are available to detect an error and the potential for misdetection is very low (1 : 2^{32}). When an error is

found, one of the syndrome bytes may be used to locate the error and another to correct the error; then only 2 syndromes are left which gives us a $1 : 2^{16}$ chance of mis-correction. The process of error correcting requires complex math in the Galois field $GF(2^8)$ but there are short cuts available in the form of two look-up tables, each 256 bytes long to evaluate powers and logarithms within the GF. All 4 syndrome bytes should be inverted at encoding and re-inverted at decoding to prevent the situation of 28 bytes of zero data giving zero parity.

[83] It is envisaged that the embodiments of the invention can achieve 1/4 of the audio sample rate because allowance is made for audio clock-slip in the decoding process. Consequently, it is possible to achieve a bit rate considerable greater than that of US2010/0161689 such that the audio WAV file in the present application can be considerably smaller. For example, for a 32Kbyte DSP/MCU, it is possible to upgrade a peripheral device or the like in under 20 seconds.

[84] In the present invention, the use of differential encoding renders the phase of the data unimportant. As such, this mitigates the prior art problem where a phase inversion in the data, i.e. a '1' appears as a '0' or vice-versa, can cause the data to be unrecognized and thus the encoding to fail.

[85] The deliberate redundancy in the symbol table of the present invention affords one or more levels of error correction which is advantageous over the known systems.

[86] In preferred embodiments of the invention, left and right stereo channels are utilized. It will be understood that more channels could be used in a multi-channel system thereby further enhancing the efficiency of the method of the invention.

[87] The synch process of the invention acts as a trigger for the DSP/MCU to automatically switch the programmable device to its upgrade mode and to stop any digital noise at the speakers as the system will mute the speakers or only play stored upgrade sound.

[88] The method of the invention is adapted to also use compressed audio formats enabling short-range wireless connections such as Bluetooth™ to be utilized instead of a cable connection.

[89] Furthermore, there is no requirement in the present invention to provide an interface unit between the digital content media player and the device being upgraded as the standard input line of a typical audio device will work and the firmware of the device being upgraded provides the interface as such.

[90] In general, the invention provides a method for modifying a system configuration on a programmable device is disclosed comprising generating a modify file in a digital media content format; storing the modify file on a digital media content player; connecting the digital media content player to the programmable device; playing the modify file with the digital media content player for transmitting data for modifying

the system configuration of the programmable device.

[91] Embodiments of the invention have been described herein, including the best mode known to the inventors for carrying out the invention. Variations of those preferred embodiments may become apparent to those of ordinary skill in the art upon reading the foregoing description. The inventors expect skilled artisans to employ such variations as appropriate, and the inventors intend for the invention to be practiced otherwise than as specifically described herein. Accordingly, this invention includes all modifications and equivalents of the subject matter recited in the claims appended hereto as permitted by the applicable law. Moreover, any combination of the above-described elements in all possible variations thereof is encompassed by the invention unless otherwise indicated herein or otherwise clearly contradicted by context.

Claims

[Claim 1] A method for modifying a system configuration of a programmable device comprising:
providing a modify file in a digital media content format to an electronic device adapted to play a digital media content file;
connecting said electronic device to the programmable device; and
playing the modify file with the electronic device for transmitting data in an analog signal format to said programmable device for modifying the system configuration of the programmable device.

[Claim 2] The method of claim 1, wherein the step of modifying the system configuration of the programmable device comprises upgrading firmware of said programmable device.

[Claim 3] The method of claim 2 or claim 3, wherein said programmable device comprises a non-network connected device.

[Claim 4] The method of any one of claims 1 to 3, wherein the electronic device is connected to the programmable device by an analog audio connection.

[Claim 5] The method of claim 4, wherein the analog audio connection is between a headphone output port of the electronic device and an analog input port or a line-in port of the programmable device.

[Claim 6] The method of any one of the preceding claims, wherein the connecting of the electronic device to the programmable device comprises connecting the electronic device to the programmable device using a stereo analog audio cable.

[Claim 7] The method of any one of the preceding claims further comprises placing the programmable device into a modifying mode prior to playing the modify file with the electronic device.

[Claim 8] The method of any one of the preceding claims, wherein the step of providing the modify file in a digital media content format comprises providing the modify file in a waveform audio format (WAVE) having a '.wav' file format extension.

[Claim 9] The method of any one of the preceding claims, wherein the step of providing the modify file comprises differential encoding an uncompressed modulated digital media content file to transmit data to the programmable device for modifying the system configuration of the programmable device.

[Claim 10] The method of any one of the preceding claims, wherein the step of

providing the modify file comprises inserting at the beginning of a data session a physical layer calibration phase to normalize data slicing thresholds and establish a baseline for zero.

[Claim 11] The method of claim 10, wherein the step of providing the modify file comprises providing hysteresis to the analog data slicing to reduce noise manifesting and jitter of digital signal output.

[Claim 12] The method of any one of the preceding claims, wherein the step of providing the modify file comprises removing low frequency content from the data spectrum of the modify file.

[Claim 13] The method of claim 12 wherein the step of providing the modify file comprises selecting channel symbols with zero DC content.

[Claim 14] The method of any one of the preceding claims, wherein the playing of the modify file with the electronic device for transmitting data for modifying the system configuration of the programmable device comprises transmitting data across two data streams of two audio channels.

[Claim 15] The method of claim 14 further comprising merging the two data streams into a final file format.

[Claim 16] The method of claim 2 wherein placing the programmable device into the modifying mode comprises muting speakers of the programmable device.

[Claim 17] An electronic system for modifying a system configuration of a programmable device, said electronic system being capable of playing a digital media content file, the electronic system comprising:
memory for storing a modify file in a digital media content format;
means for connecting to a programmable device; and
means for playing the modify file and formatting data of said modify file to an analog signal format to transmit said data in said analog signal format to said programmable device.

[Claim 18] A method of formatting a digital media content file into an audio signal format comprising: converting data of said media content file from an m bit code space to respective n bit channel symbols, where m and n are positive integers, $m < n$ and the channel symbols are chosen from an available set of channel symbols to have zero DC content.

[Claim 19] The method of claim 18, wherein $m = 4$ and $n = 6$ to provide a four-to-six modulation scheme.

[Claim 20] A computer readable medium comprising program code executable by a processor of an electronic device to implement the steps of the

methods of any one of claims 1 to 16 or 18 and 19.

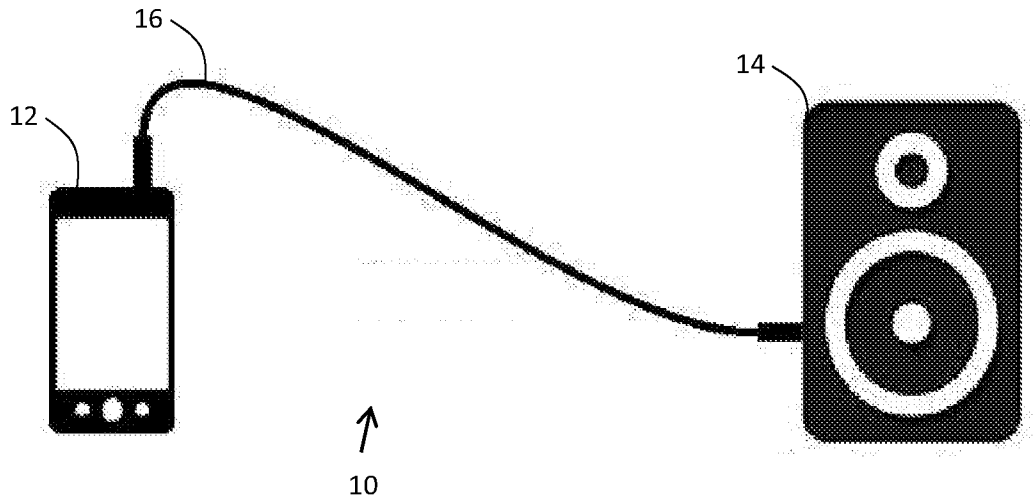

16
14
12
10

FIG. 1

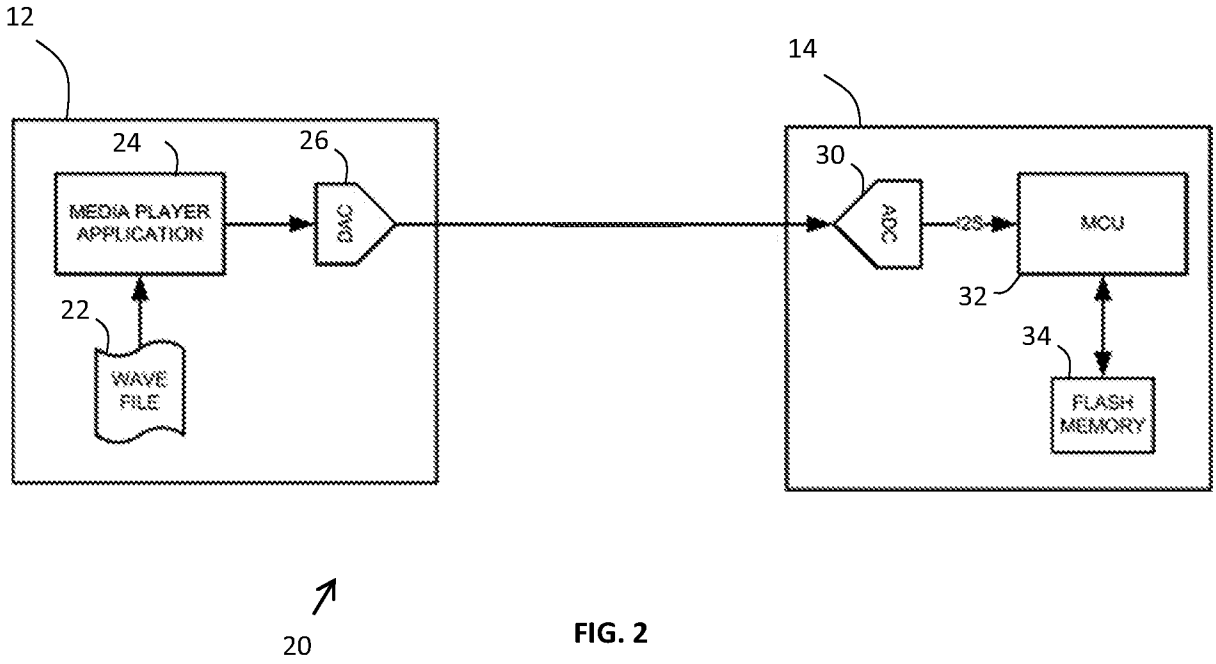

12
14
24
26
30
MEDIA PLAYER APPLICATION
DAC
ADC
I2S
MCU
22
32
34
WAVE FILE
FLASH MEMORY
20

FIG. 2

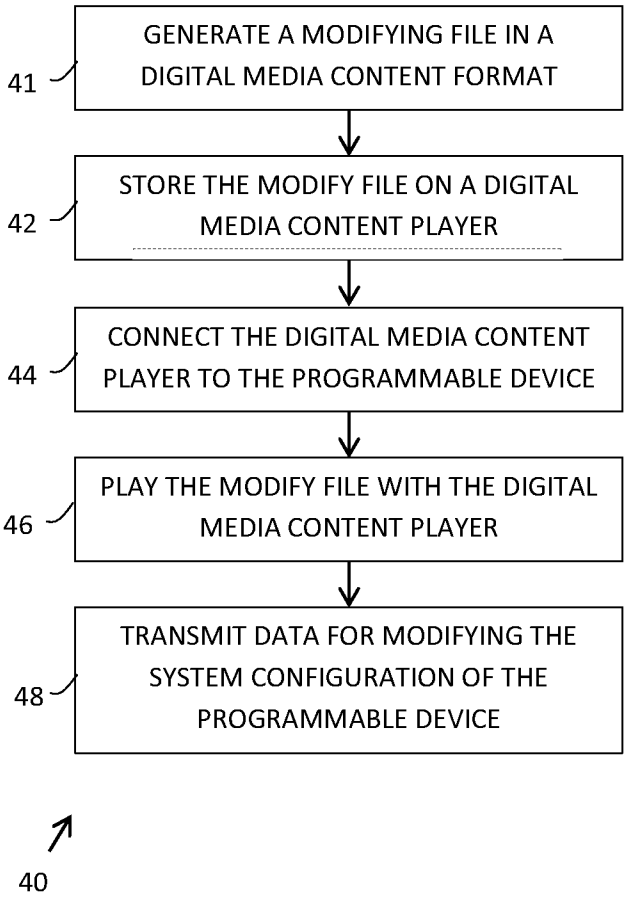

GENERATE A MODIFYING FILE IN A DIGITAL MEDIA CONTENT FORMAT
41
STORE THE MODIFY FILE ON A DIGITAL MEDIA CONTENT PLAYER
42
CONNECT THE DIGITAL MEDIA CONTENT PLAYER TO THE PROGRAMMABLE DEVICE
44
PLAY THE MODIFY FILE WITH THE DIGITAL MEDIA CONTENT PLAYER
46
TRANSMIT DATA FOR MODIFYING THE SYSTEM CONFIGURATION OF THE PROGRAMMABLE DEVICE
48
40

FIG. 3

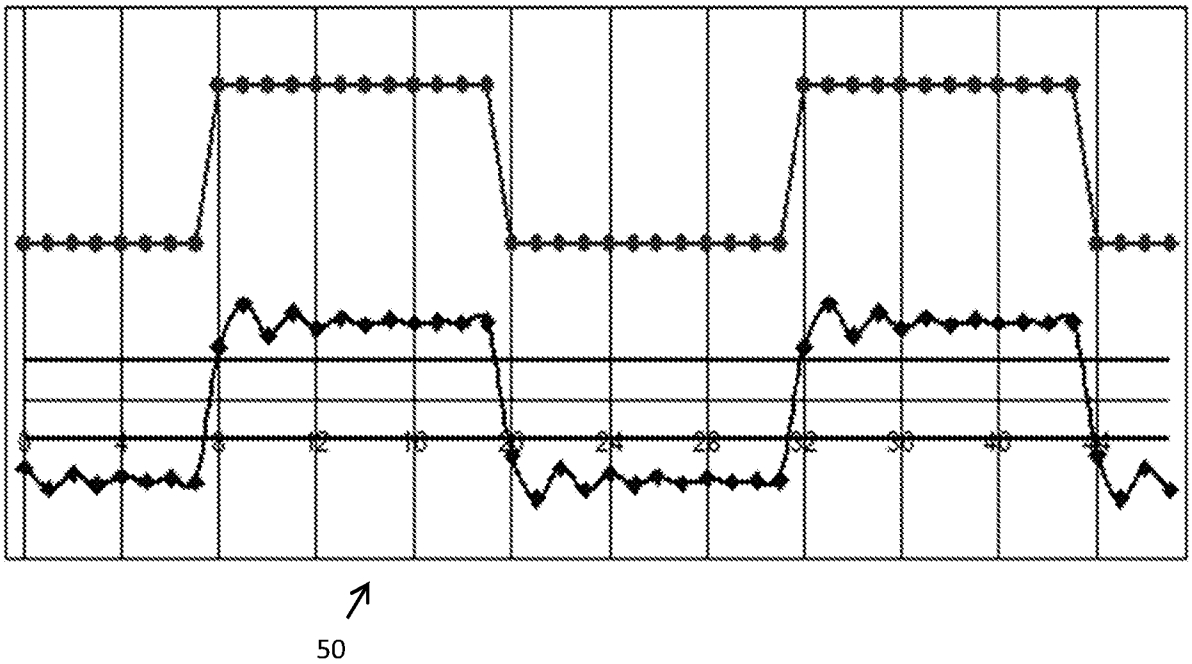

50

FIG. 4

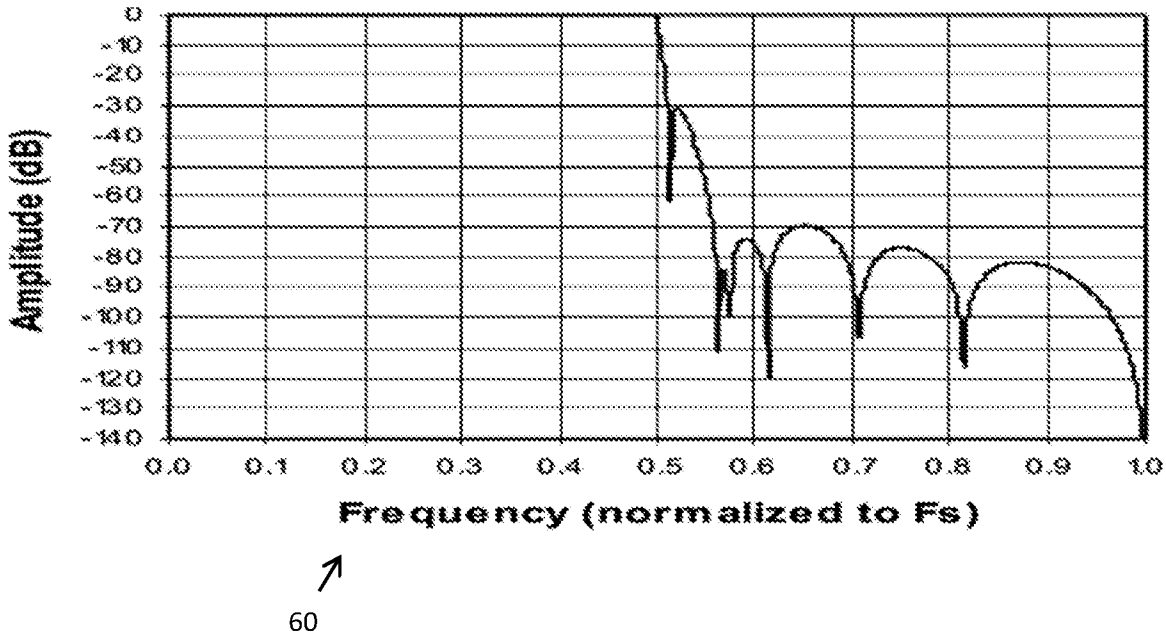


FIG. 5

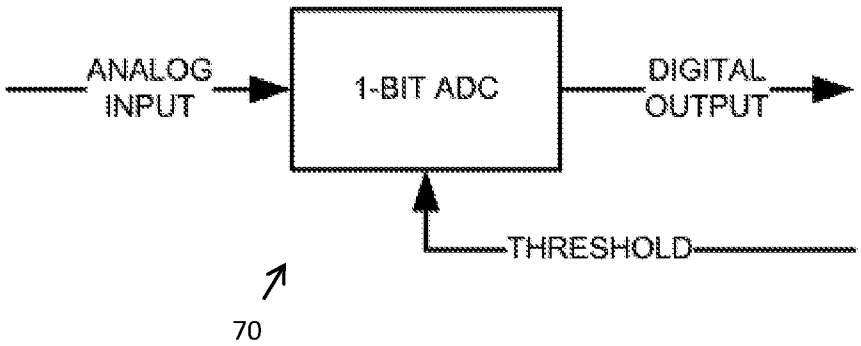


FIG. 6

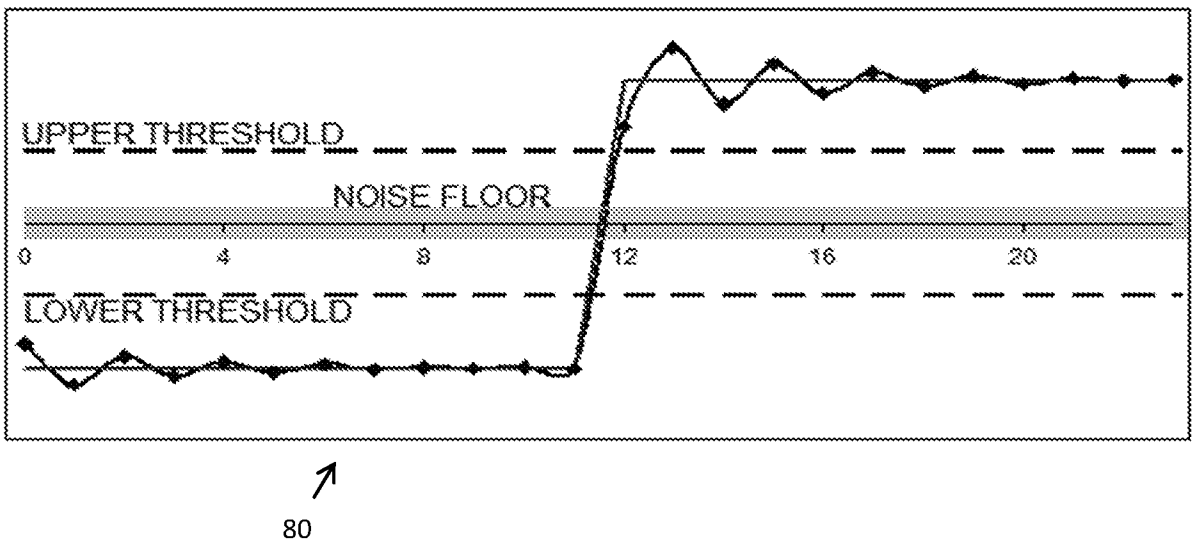
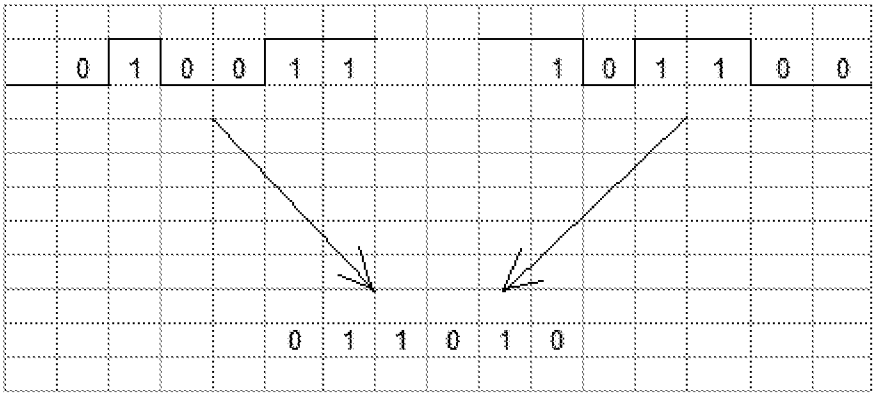


FIG. 7

Encoder			Decoder		
Y[n-1]	X[n]	Y[n]	X[n-1]	X[n]	Y[n]
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0

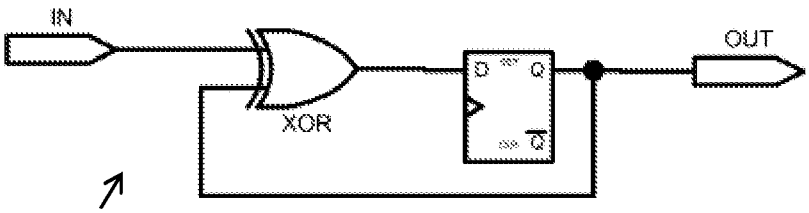
90

FIG. 8



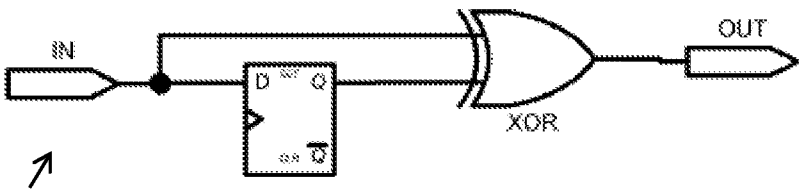
100

FIG. 9



110

FIG. 10



120

FIG. 11

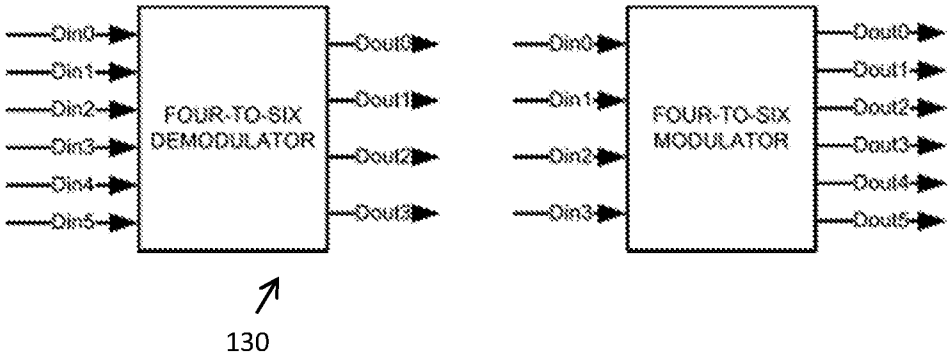
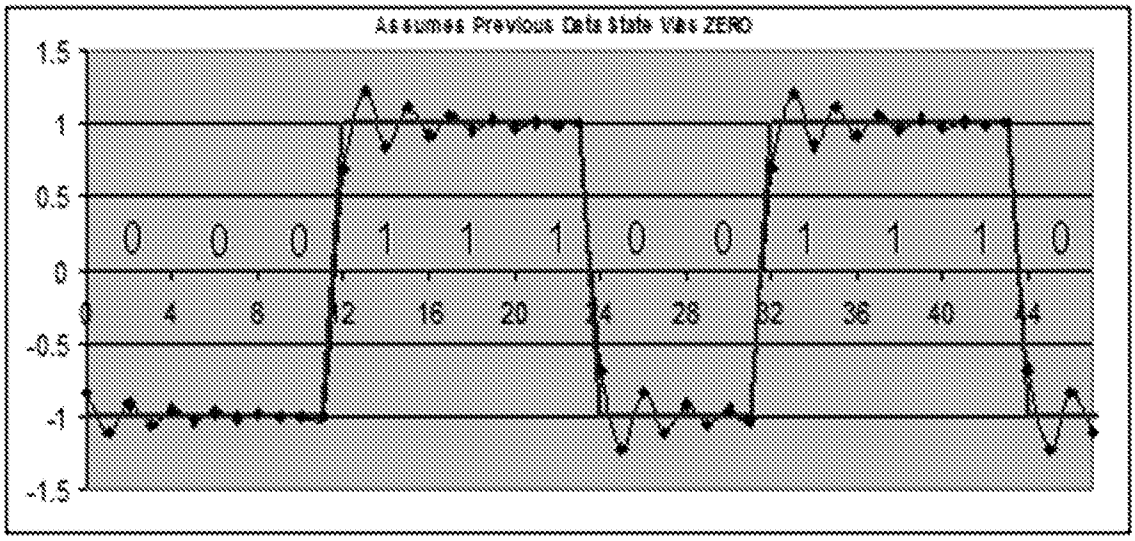


FIG. 12

Data (Binary)	Data (4 bit Hex)	FSM Channel Symbol Hex (6 bit Hex)	FSM Channel Symbol (Binary)
0000	0x0	0x04	000100
0001	0x1	0x09	001001
0010	0x2	0x0B	001011
0011	0x3	0x0E	001110
0100	0x4	0x12	010010
0101	0x5	0x15	010101
0110	0x6	0x17	010111
0111	0x7	0x1A	011010
1000	0x8	0x1D	011101
1001	0x9	0x1F	011111
1010	0xA	0x24	100100
1011	0xB	0x29	101001
1100	0xC	0x2B	101011
1101	0xD	0x2E	101110
1110	0xE	0x32	110010
1111	0xF	0x35	110101

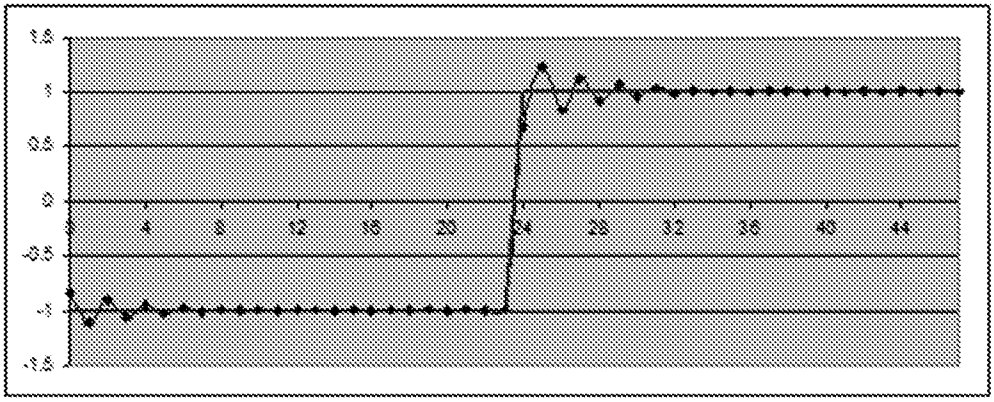
140

FIG. 13



150

FIG. 14



160

FIG. 15

Channel Code (NAN)	Channel Symbol (6 bit)
W	0x37
X	0x3A
Y	0x3D
Z	0x3F

170

FIG. 16

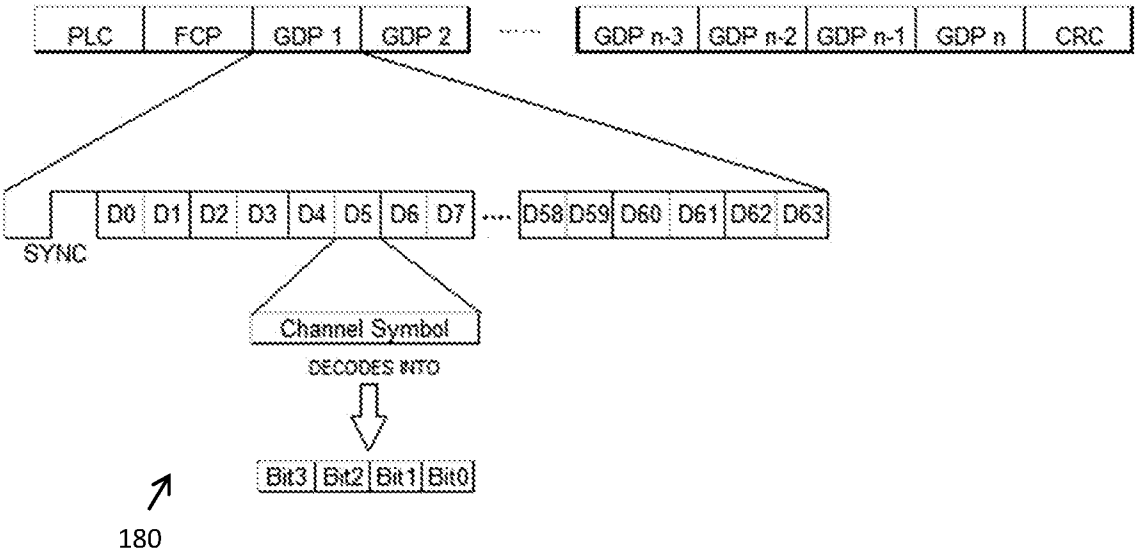


FIG. 17

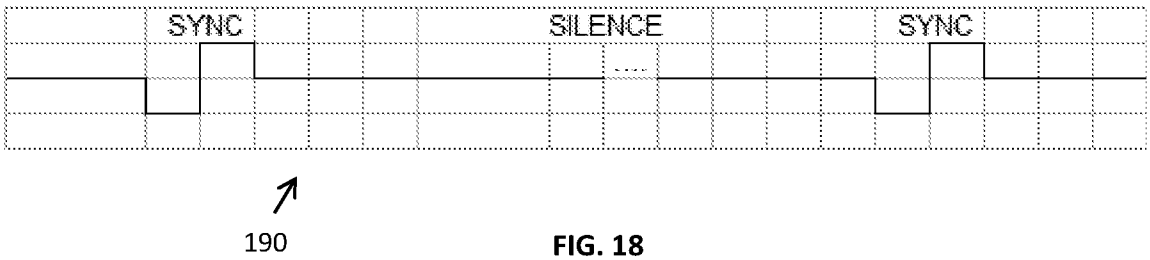


FIG. 18

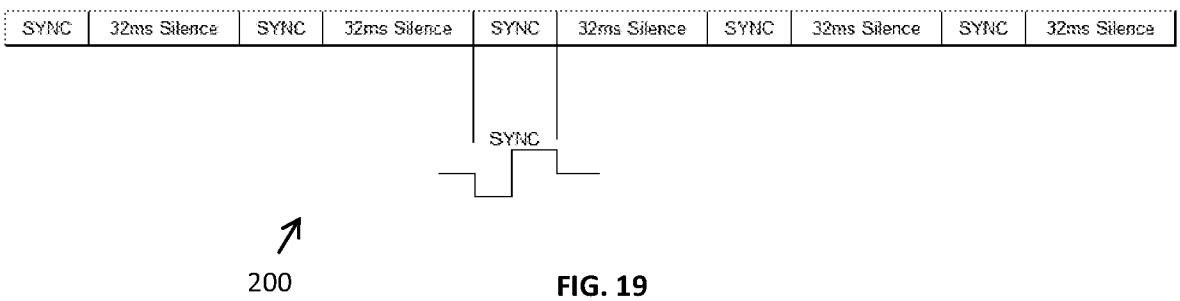


FIG. 19

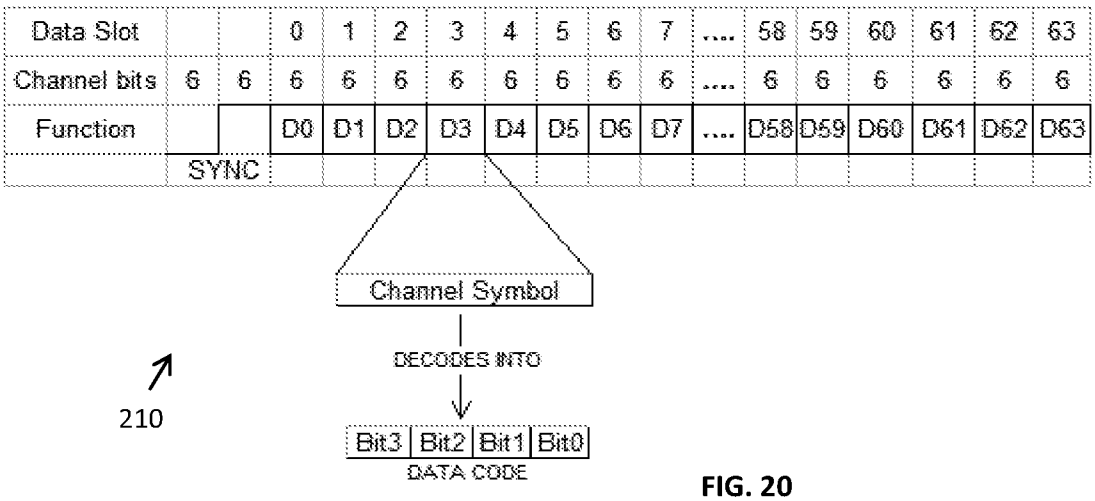
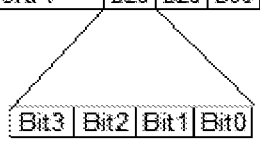


FIG. 20

	MSB			LSB			MSB			LSB
Word 0	D0	D1	D2	D3		Word 8	D32	D33	D34	D35
Word 1	D4	D5	D6	D7		Word 9	D36	D37	D38	D39
Word 2	D8	D9	D10	D11		Word 10	D40	D41	D42	D43
Word 3	D12	D13	D14	D15		Word 11	D44	D45	D46	D47
Word 4	D16	D17	D18	D19		Word 12	D48	D49	D50	D51
Word 5	D20	D21	D22	D23		Word 13	D52	D53	D54	D55
Word 6	D24	D25	D26	D27		Word 14	D56	D57	D58	D59
Word 7	D28	D29	D30	D31		Word 15	D60	D61	D62	D63



220

FIG. 21

Data Codes (4 bits per code)	Name	Size/Function
D0	T[3:0]	4 bits File type: 0001b = Firmware upgrade file 0010b = Audio sample upgrade file All other types to be defined
D1 to D5	L[23:0]	24 bits Length in bytes of the attached file. MSB first
D6 to D9	P[15:0]	16 bits Product code
D10 to D13	V[15:0]	16 bits Version number, in the case of firmware upgrade, this is the version of the firmware included in the manifest and can be used to prevent from overwriting with older firmware versions
D14 to D59	x	Undefined
D60 to D63	C[15:0]	16 bits CRC of the FCP D[0:59]

230

FIG. 22

D0	T[3:0]	L[23:20]	L[19:16]	L15:12]
	L11:8]	L[7:4]	L[3:0]	P[15:12]
	P[11:8]	P[7:4]	P[3:0]	V[15:12]
	V[11:8]	V[7:4]	V[3:0]	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	x	x	x	x
	C[15:12]	C[11:8]	C[7:4]	C[3:0]

D63

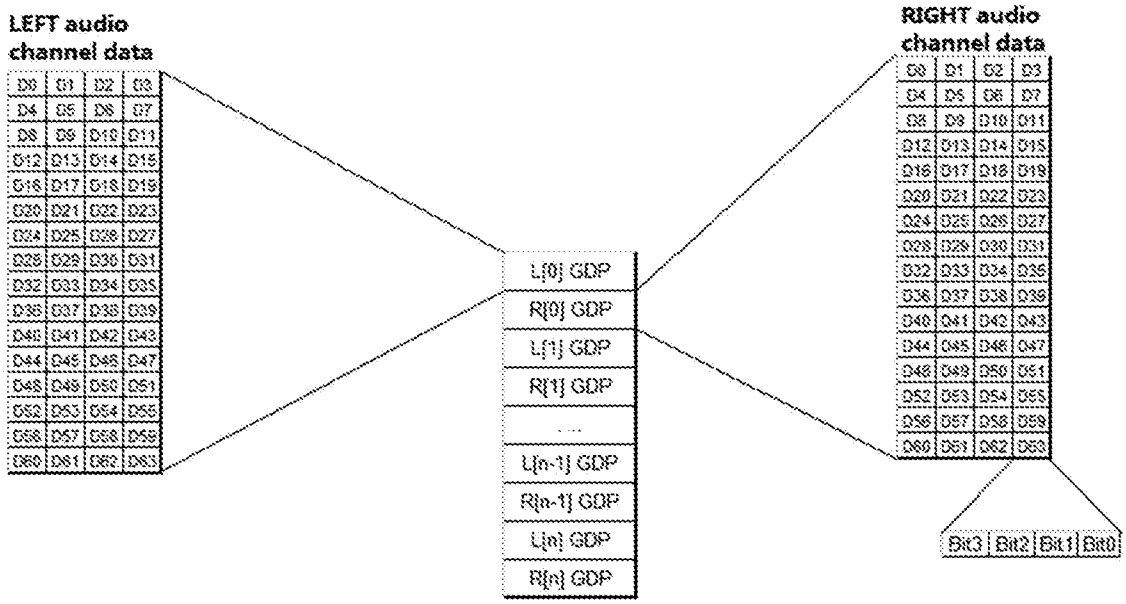
240

FIG. 23

Data Codes (4 bits per code)	Name	Size/Function
D0 to D7	CRC[31:0]	32 bits CRC
D8 to D63	x	Undefined

250

FIG. 24



260

FIG. 25

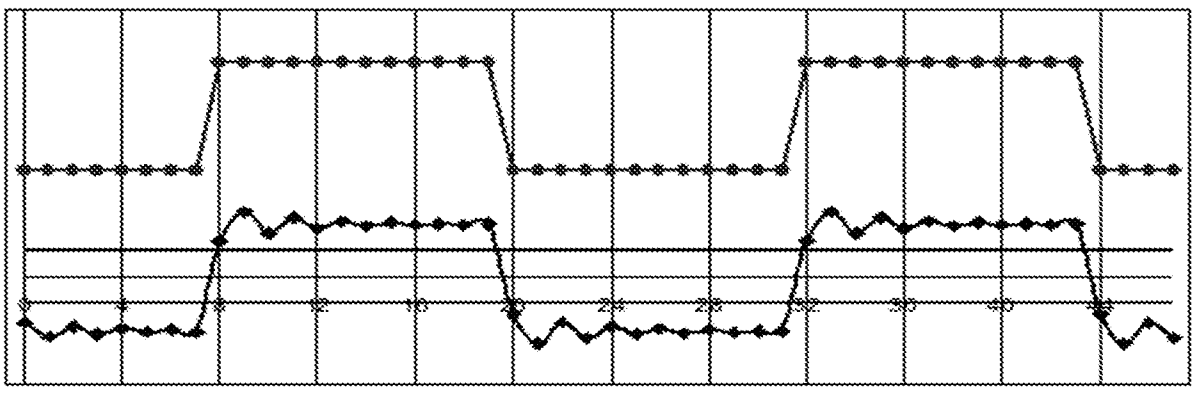
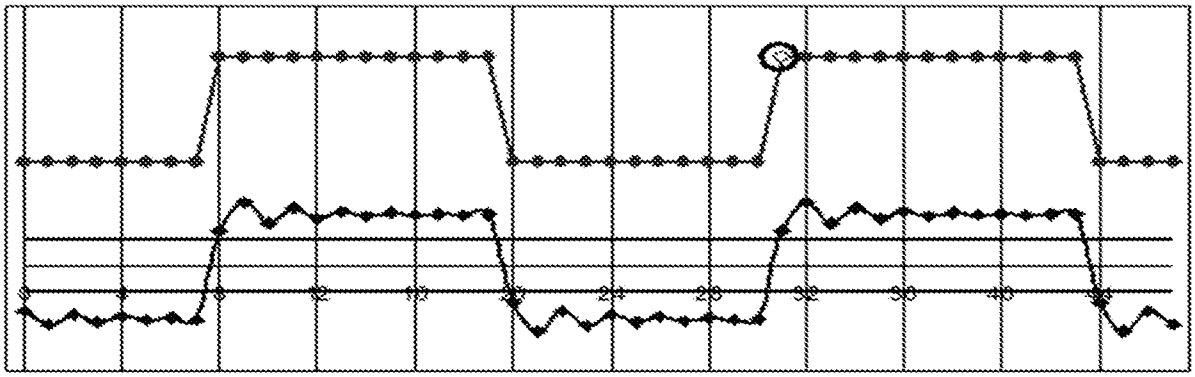


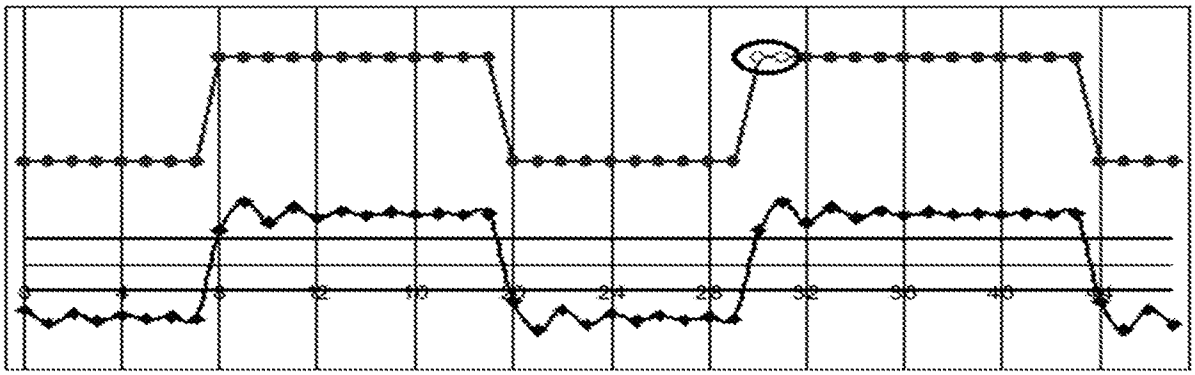
FIG. 26

270



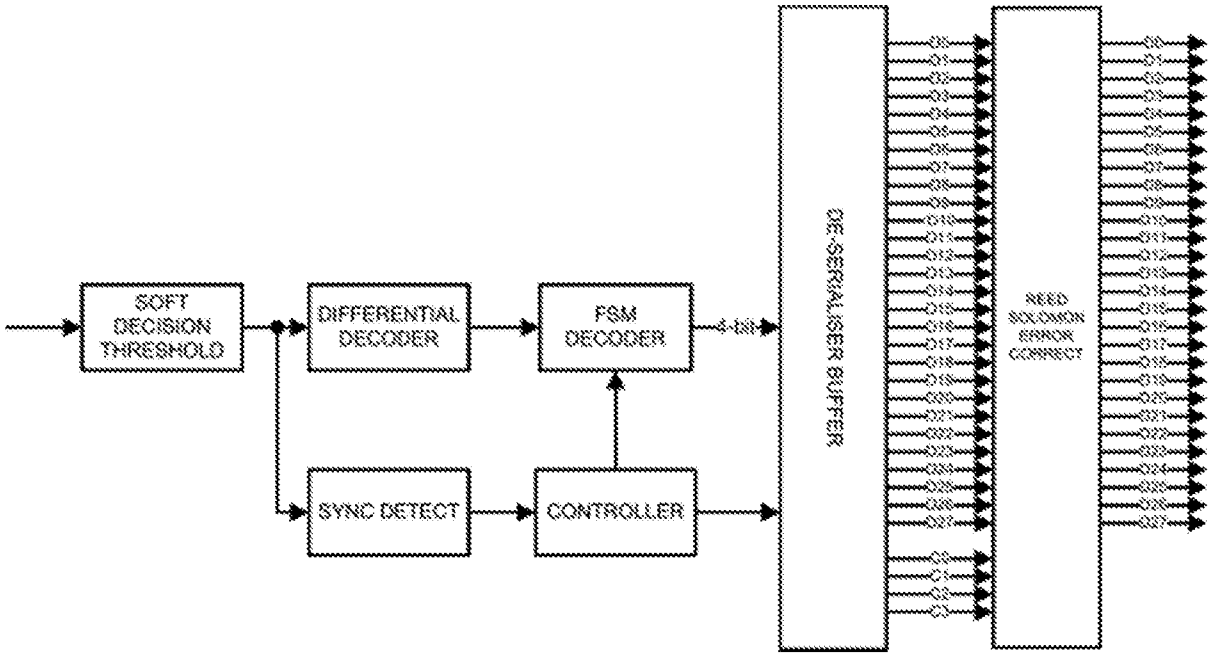
280

FIG. 27



290

FIG. 28


SOFT DECISION THRESHOLD
DIFFERENTIAL DECODER
FSM DECODER
4-bit
SYNC DETECT
CONTROLLER
DE-SERIALISER BUFFER
REED SOLOMON ERROR CORRECT

300

FIG. 29

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2015/072263

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/445(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/-

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNMED;CPRSABS;CNABS;CPEA;TWMED;KRABS;DWPI;JPABS;ILABS;RUABS;TWABS;HKABS;MOABS;DEABS; SIPOABS;SGABS;AUABS:upgrad+, file?, media, updat+, analog, modify+, audio, digital, video

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 101038553 A (SUOJIE SCI & TECHNOLOGY CO., LTD.) 19 September 2007 (2007-09-19) claims 1-14, description, page 2, line 1-page 5, line 10 and figures 2 and 3	1-3, 7-13, 16, 17, 20
Y	CN 101038553 A (SUOJIE SCI & TECHNOLOGY CO., LTD.) 19 September 2007 (2007-09-19) claims 1-14, description, page 2, line 1-page 5, line 10 and figures 2 and 3	4-6, 14, 15
Y	US 2011088024 A1 (TACKETT ET AL.) 14 April 2011 (2011-04-14) description, paragraphs [0023]-[0040], [0067] and figures 1 and 2	4-6, 14, 15
X	US 4626826 A (SONY CORP.) 02 December 1986 (1986-12-02) claim 1 and description, column 1, line 15-column 5, line 51	18-20
A	CN 104321745 A (TELERIK AD) 28 January 2015 (2015-01-28) the whole document	1-20

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 October 2015	**29 October 2015**
Name and mailing address of the ISA/CN **STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China** Facsimile No. **(86-10)62019451**	Authorized officer **WANG,Xiaowei** Telephone No. **(86-10)62411642**

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2015/072263

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	101038553	A	19 September 2007	None			
US	2011088024	A1	14 April 2011	KR	20120091201	A	17 August 2012
				WO	2011047152	A3	24 November 2011
				EP	2488945	A2	22 August 2012
				WO	2011047152	A2	21 April 2011
				US	8739149	B2	27 May 2014
				CN	102576306	A	11 July 2012
US	4626826	A	02 December 1986	EP	0143553	A1	05 June 1985
				CA	1244950	A1	15 November 1988
				JP	S6093857	A	25 May 1985
				DE	3475978	D1	09 February 1989
				EP	0143553	B1	04 January 1989
				JP	H0683271	B2	19 October 1994
CN	104321745	A	28 January 2015	AU	2013217338	A1	21 August 2014
				EP	2812797	A2	17 December 2014
				JP	2015510635	A	09 April 2015
				US	2013205277	A1	08 August 2013
				WO	2013117995	A2	15 August 2013
				WO	2013117995	A3	31 October 2013
				CA	2863882	A1	15 August 2013
				KR	20150047453	A	04 May 2015
				SG	11201404680	A1	26 September 2014