PROVISIONAL PATENT APPLICATION

for

UNITED STATES LETTERS PATENT

on

INTELLIGENT LED PIXEL ILLUMINATION SYSTEM

INVENTORS: LEO M. FERNEKES of Berkeley, CA

APPLICANT/ASSIGNEE: Ultralis Ltd.
Flat D, 32/F, Block 1, Central Park
18 Hoi Ting Road
Mong Kok, Kowloon, Hong Kong

ATTORNEY DOCKET NO.: COH-00625

SHEETS OF DRAWINGS: EIGHT (8)

INTELLIGENT LED PIXEL ILLUMINATION SYSTEM

FIELD OF THE INVENTION

[0001] The present invention relates generally to illumination systems and, more particularly, to light emitted diode (LED) illumination systems.

BACKGROUND OF THE INVENTION

[0002] LEDs are widely used for illuminating signage, point of purchase displays and many other lighting applications. The availability of full color RGB (Red Green Blue) LEDs allows creating attractive animated signage by dynamically controlling the intensity of the LEDs on an individual basis.

[0003] The typical methods employed to control arrays of RGB LEDs in a signage application require either: (a) complex custom wiring, requiring significant labor to install and use, (b) inflexible custom display units designed for a specific application which are only cost-effective in large volume, (c) inflexible dot-matrix displays which do not lend themselves to designs deviating from a fixed grid layout. The solutions currently employed require significant effort to create and map the visual content to the correct pixels.

[0004] The invention provides a method to create any arbitrary arrangement of pixel elements, capable of displaying coordinated animations and lighting effects without the need for any custom wiring, circuit boards, or other labor-intensive fabrication methods. The task of pixel mapping and content creation is also significantly simplified, creating opportunities for low-volume customization that were not previously cost effective.

SUMMARY OF THE INVENTION

[0005] Briefly, and in general terms, the invention provides an illumination system having a plurality of LED Pixel elements. The LED Pixels can comprise programmable memory, logic, LED’s, and a light sensor.

[0006] More specifically, in an exemplary embodiment, by way of example and not limitation, pixel elements are electrically connected in parallel and driven by a circuit that provides power, synchronization, and command signals via a 2-pole circuit, by reversing the polarity of the applied voltage. LED Pixels (hereinafter LP) can be installed on an insulating substrate clad on both sides with a conductive layer, providing a 2-circuit conductive path to each pixel, allowing random placement without concern for wiring, each Pixel is secured into a drilled hole with a fastener ensuring contact with both conductive layers.

[0007] LED Pixels are programmed with visual content by optical means, by illuminating each pixel with binary coded light pulses while clock signals are applied electrically. This mechanism allows programming the pixels in the final assembled configuration. In normal operation, the LED Pixels synchronously display sequential 'frames' of visual content at a rate controlled by the driver, creating a smooth and continuous visual output.

[0008] For purposes of summarizing the invention and the advantages achieved over the prior art, certain advantages of the invention have been described herein. Of course, it is to be understood that not necessarily all such advantages may be achieved in accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention may be embodied or carried out in a manner that achieves or optimizes one advantage or group of advantages as taught herein without necessarily achieving other advantages as may be taught or suggested herein.

[0009] All of these embodiments are intended to be within the scope of the invention herein disclosed. These and other embodiments of the present invention will become readily apparent to those skilled in the art from the following detailed description of the preferred embodiments having reference to the attached figures, the invention not being limited to any particular preferred embodiment disclosed

BRIEF DESCRIPTION OF THE DRAWINGS:

[0010] Fig. 1 is an internal block-diagram of an exemplary embodiment of an LED Pixel (LP) unit in accordance with the invention.

[0011] Fig. 2 is an internal block-diagram of the LED Pixel Driver unit of the LED Pixel unit of FIG. 1.

[0012] Fig. 3 is a block-diagram showing an exemplary embodiment of a method of programming the LED Pixels using a video projector in accordance with the invention.

[0013] Fig. 4 is a block-diagram showing multiple LED pixels in a normal deployed configuration in accordance with the invention.

[0014] Fig. 5 is a block diagram of a stand-alone programing device for the LED Pixels in accordance with the invention.

[0015] Figs. 6A-6D are timing diagrams showing four different modes of operation in accordance with the invention, including run mode, command mode, read frames mode, and programming mode.

[0016] Fig. 7 is a simplified map of a projected raster of the video projector, used for programming LED Pixels, in accordance with the invention.

[0017] Fig. 8 is an illustration showing exemplary LED Pixels of FIG. 1, mechanically mounted in a substrate.

[0018] Fig. 9 is a flowchart showing an exemplary logical flow of START MODE in accordance with the invention.

[0019] Fig. 10 is a flowchart showing an exemplary logical flow of SELF-TEST MODE in accordance with the invention.

[0020] Fig. 11 is a flowchart showing an exemplary logical flow of PROGRAM MODE in accordance with the invention.

[0021] Fig. 12 is a flowchart showing an exemplary logical flow of RUN MODE in accordance with the invention.

[0022] Fig. 13 is a flowchart showing an exemplary logical flow of SEND FRAMES MODE in accordance with the invention.

[0023] Fig. 14A is an exemplary flowchart showing sign design and manufacture steps- using in-place programming concept in accordance with the invention.

[0024] Fig. 14B is an exemplary flowchart showing sign design and manufacture steps- using the external programming concept in accordance with the invention.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0025] With reference to the drawings, there is shown a "2-terminal" electrically powered light emitting device, LED Pixel (LP), containing three separate color emitting elements; Red, Green, Blue, along with drive circuitry, logic, reprogrammable FLASH memory, and a single light sensitive element. The light emitting and sensing elements are positioned in close proximity to each other with their active surfaces facing generally perpendicular to the mounting substrate.

[0026] In a detailed aspect of an exemplary embodiment , a plurality of LPs are mechanically fastened to a substrate having perforated sheets of material comprising laminated layers of conductive and non-conductive materials, providing an isolated 2-circuit conductive medium to connect all LPs in parallel and to provide mechanical registration and fixation. In an exemplary embodiment, as discussed in detail with reference to FIG. 8. Threaded fasteners can be used to both affix and electrically connect the LPs to the substrate. A first circuit of the 2-circuits can be electrically connected via an exposed metallic housing of the LP making contact with a front conductive surface of the substrate. The second circuit of the 2-circuits can be electronically connect via the screw in contact with a rear conductive surface of the substrate, providing a conductive path to an inner isolated female threaded contact inside the LP. Other means of assembly may be employed including interference fits, spring clips, push nuts, adhesives, spot welding, etc.

[0027] In the exemplary embodiment, each LP is capable of storing and sequentially displaying a large number of preset programmable RGB light values, synchronous with an externally generated clock signal. When combined in parallel-connected arrays, LPs can serve as individual pixels in a dot matrix display array, each one displaying a different light intensity and color corresponding values in the overall displayed image. Each image is advanced simultaneously by the external clock, causing a plurality of LPs to synchronously display 'frames' of visual information. By displaying these frames in sequence at a sufficiently high rate, the displayed information can take the form of moving video images, motion graphics, numeric, and alphanumeric data and coordinated dynamic lighting effects.

[0028] Since the LP is electrically a 2-terminal device, its power and synchronization must be delivered over the same two input terminals. In a preferred embodiment, synchronization signals are delivered to the LP by rapidly reversing the polarity of the voltage applied to the LP, an internal rectifier bridge ensures that the internal electronics are supplied with the correct voltage polarity regardless of the applied input polarity. A separate input connection ahead of the bridge rectifier is used to determine the absolute polarity of the applied external voltage. An internal capacitor is utilized as a local reservoir of energy to allow the LP's circuitry to receive uninterrupted power during the switching interval. An ideal drive is a square wave that reverses polarity rapidly, but with a controlled edge slew rate to minimize the generation of EMI during switching transitions.

[0029] In the exemplary embodiment, the connection wiring polarity of the LP is non-critical due to the fact that the LP is only sensitive to input signal edge transition timing, not the absolute polarity of the applied voltage.

[0030] Overall synchronization and control is accomplished by pulse width coded drive signals. In the normal mode of operation, frames of visual information are displayed sequentially by advancing the frame data pointer in each LP when the input power changes polarity, this allows for displaying video at a variable frame rate controlled by the driver. A master reset signal is generated by sending a pulse that is significantly shorter than the maximum frame rate to all the LPs, causing them all to reset to frame #1 internally, ensuring that all LPs are synchronized.

[0031] Power cycling the LP by applying 0 volts to the LP for a minimum time period to allows the internal circuitry to reset, placing the LP in a known state. Additional sequences of coded pulses are used to communicate special status and control commands to all connected LPs.

[0032] LPs are programmable devices; the internal flash memory can be reprogrammed to change the displayed images.

[0033] The LP utilizes an integrated RGB LED and driver IC containing a current regulator and pulse width modulator in one package. This device communicates with the LP microcontroller via a synchronous serial bus.

[0034] The LPs utilize an internal light sensor to encode new sequences of frame data, the data is encoded as simple digital information, represented by optical light or dark signals focused on the face of each LP.

[0035] The LPs are placed into programming mode by sending a command (e.g., pulse width coded command); the data is then sent optically to individual LPs while the clock is applied globally to all connected LPs via the electrical input. This system allows gang programming a group of LPs simultaneously with new content.

<u>Fig. 1 - Architecture of the LED Pixel</u>

[0036] With reference now to FIG. 1, an exemplary embodiment of the LP is shown, which utilizes an Application Specific Integrated Circuit (ASIC) to implement all the main functions of the LP, the current embodiment utilizes a micro-controller and other discrete components to accomplish these desired functions.

[0037] Power is applied to the LP via the two input terminals 100, illustrated as A,B.
The polarity of the applied voltage reverses periodically during operation, the rectification circuit 101 provides a steady DC voltage to power the internal circuitry of the LP without interruption, a capacitor 102 providing a reservoir of energy to sustain the voltage through the brief polarity reversal cycle. A polarity detector circuit 103 connected to the input, ahead of the rectification circuit provides a logic level corresponding to the input polarity to the Logic and

Memory circuit for synchronization and control. The core functions of the LP are controlled by the Logic and Memory Circuit 104, which controls recording and playback of visual frame sequences.

[0038] The LED Driver circuit 108 comprises three channels, (R,G,B). Each channel includes a constant current regulator 109, pulse-width-modulator 107, and driving switching element 111 that determines the brightness of each LED 110 independently. In a preferred embodiment, the LP contains three semiconductor LED chips, each producing a different wavelength of light, (Red, Green, Blue) these chips provide the final optical output of the device.

[0039] A light sensing element 106 is employed to receive optically transmitted digital signals from external programming devices. In exemplary embodiments, this is typically a photo-diode, photo-transistor, or one of the three LEDs utilized in a photo-conductive mode, among others. It is desirable to mount the light sensing element in a manner to protect it from light entering from a generally non-axial direction, to avoid corruption of the received data during programming. In other embodiments, an additional electrical contact can be used to electrically convey programming data at a greater speed.

Fig.2 - Architecture of the Driver

[0040] With reference now to FIG. 2, an exemplary architecture of the LED Driver 108 is shown. A DC power source 200 supplies power to a MOSFET H-Bridge switch 201, capable of switching many amperes with minimal voltage losses. The H-Bridge switch is capable of reversing the polarity of voltage at the output terminals 202. The action of this H-bridge is controlled by a micro-controller (MCU) 204. A current sensor such a current transformer or hall sensor 203 measures the current flow in the output circuit and delivers this value as an analog signal to the MCU for the purpose of decoding communications from the LPs, connected to the two output terminals A,B. An additional sync input 205 can accept signals from other devices to control and synchronize the playback of content.

[0041] In a detailed aspect of an exemplary embodiment, the complete system requires the driver to generate a correctly coded sequence of polarity reversal pulses to drive an array of parallel-connected LPs. In normal operation, the driver sends a repeating string of N pulses to the LPs equal to the number of frames in the visual sequence, followed by a frame reset pulse to reset the LPs to frame #1. The rate that these pulses are generated controls the overall frame rate of the displayed content. For this system to function correctly, the driver must know the number of frames programmed into the LPs, in practice, this information is conveyed at startup by the mechanism described below.

[0042] The driver utilizes current sensor 203 to measure the current flow through the LP array, this sensor forms a communication channel that allows the LPs to collectively communicate data in the reverse direction, from LPs to driver, by turning the LPs LEDs on and off in response to queries from the driver. At startup, the driver will request the number of frames stored in the LP array, the array responds with a string of digital data coded as high or low current demand, by pulsing the LEDs on or off, changing the level of current flowing in the power circuit. The current in the circuit is substantially higher when the LPs turn on their LEDs, so it is very easy for the driver to differentiate the two states.

Light sensor programming operation:

[0043] The light sensor system used in programming must cope with a range of light levels. Once the LP receives a command to enter programming mode, the light sensing system enters calibration mode where the LP expects eight alternating light and dark pulses, corresponding to the one or 0 state, these pulses are used to calibrate the light sensing threshold of the sensor, to allow the LP to cleanly differentiate the two states, 1 or 0. If the light sensor does not detect this calibration sequence at a sufficient brightness and contrast ratio, the programming cycle is aborted, this mechanism allows parallel-connected LPs to be selectively programmed by shielding them from light, in which case, the programming cycle is aborted, and the existing data in memory is preserved.

[0044] In a preferred embodiment, the light sensor is a PIN Photodiode connected in parallel with a small capacitor, the capacitor is charged to 5 VDC prior to the light measuring cycle, when the cycle is initiated, the charging source is disconnected and the photo current begins to discharge the capacitor at a rate proportional to the light intensity. The voltage on the capacitor is measured against a threshold, when the threshold is reached, the measurement cycle is complete. The time required for the voltage to reach the threshold is inversely proportional to the light level and is measured by a gated digital counter driven by a high-speed clock, this counter is typically 16 bits, providing high-resolution measurements. If the sensor is not exposed to light, the measurement cycle may lapse before the threshold is reached, in this case, the measurement is considered a 0 level.

[0045] The action of the light sensor serves to integrate the light exposure over the measurement period, making the system compatible with 'noisy' light sources, such as a DLP video projector, which has a rotating color wheel that introduces many optical pulses within a single frame display cycle.

Video Projector as programming device.

[0046] One method of programming the LP with visual content is to utilize a video projector, the video projector can provide a multiple steams of coded light pulse data to many LPs simultaneously.

[0047] The programming sequence is performed at a constant rate, in the case of a video projector, this rate is the native fame rate of the projector, which is typically 60 Hz. A video is created that codes specific regions of screen pixels either light or dark, corresponding to binary 1 or 0. These screen regions are mapped to physically correspond to the matrix of LPs to be programmed by the projector light beam. A small region of the screen is coded to produce an alternating light-dark-light-dark sequence during the entire programming video to act as a synchronizing clock signal for the programming process. A separate light sensing element is placed at this point to detect and convert this alternating clock signal into electrical pulses that are fed to the driver, which replicates the clock as polarity reversal pulses to clock the LPs directly. Thus, the LPs are provided a source of both clock and data, by electrical and optical means simultaneously.

Examples of communication and operations of the LP system.

Normal playback:

[0048] Power on reset -> Driver applies static power to LP to ensure all LP's are ready. (100 Ms) -> Driver sends 8 bit MODE command requesting the number of frames in that are stored in the LP's -> Driver sends 16 clock pulses, the LP's respond by returning the frame count as: idle current = 0, Full current =1. -> Driver begins sending frame advance pulses at the predetermined frame display rate, clocking the display of content -> Driver counts to frame 'N' then sends short frame reset signal resetting the pins to frame 1. (Cycle repeats)

Program Mode

[0049] Power on reset -> Driver applies static power to LP to ensure all LP's are ready. (100 Ms) -> Driver sends 8 bit MODE command setting PROGRAM MODE -> Driver switches its clock input to the external clock light sensor-> the LP's wait for the first rising edge of the clock signal- from the light sensor -> first light pulse from projector starts the calibration sequence of 8 alternating pulses -> 8 pulses are measured and recorded -> the LP calculates the average of both the 4 light and 4 dark pulses to determine if the signal meets minimum specifications, a threshold is established for subsequent data recovery -> the driver sends the number of frames to be recorded as a 16 bit pulse coded sequence -> the driver sends (N frames *24 bits) pulses to the LP while specific data is sent optically to each LP being programmed -> once N frames have been sent, an 8 bit checksum is sent to validate the integrity of the data -> if the checksum matches, the LP indicates a successful programming sequence by illuminating its internal GREEN LED.

Fig. 3 - Video Projector Programming

[0050] With reference now to FIG. 3, a video projector 300 can be used for programming an LP sign 308 with visual content. The sign 308 includes a plurality of LPs 310. The video projector 300 plays a specially coded video file 302 via a video file server / player 301. The projected raster 306 covers the entire sign 308 and all LPs 310 on the sign. The video projector 300 then provides multiple steams of coded light pulse data to all LPs 310 simultaneously. The programming sequence is performed at a constant rate, in the case of a video projector; this rate is the native fame rate of the projector, which is typically 60 Hz. A video is created that codes specific regions of screen pixels either light or dark, corresponding to binary 1 or 0. These screen regions 700 are mapped to physically correspond to the matrix of LPs to be programmed by the projector light beam. A small region of the screen 701 is coded to produce an alternating light-dark-light-dark sequence during the entire programming video to act as a synchronizing clock signal for the driver 304. A separate light sensing element 307 is placed at this point to detect and convert this alternating clock signal into electrical pulses that are fed to the driver 304, which replicates the clock as polarity reversal pulses to clock the LPs directly. The Driver provides LP array with power and clock signals via 2-wire connection 309, while data for each individual LP is delivered optically via individually coded pixel regions in the video file. Other means such as beam-steered lasers or other spatial-light-modulation techniques may also be employed to rapidly program the LP's.

Fig. 4 - LP and driver in a normal signage application

[0051] An array of LED pixels 400 are shown connected in parallel via conductive circuit path 401 and connected to the output terminals 402 of the driver 403, which is powered by a source of DC 404. An external sync input 405 may be used to synchronize multiple driver units for large-scale installations.

Fig. 5 - Stand-Alone programming fixture

[0052] An alternate programming system allows LPs to be programmed before installation into an application, this device programs groups of LPs loaded into the programmer in removable trays.

[0053] The stand-alone fixture comprises a box with a compartment that receives a removable tray containing multiple LPs, contained in individual numbered compartments.

[0054] A tray 501 loaded with LPs is inserted into the programmer, electrical contacts in the tray connect to the 2 connection points on each LP, connecting them in parallel to the power driver 500, the power driver is similar to the driver circuit shown in fig 2, and is powered by a

DC power source 508. An array of visible or infrared emitting LED's 502 physically corresponding to each LP in the tray 501 are utilized to transmit the optical data to each LP independently, under control of a built in micro-controller unit 505, via LED driver circuit 503. The data file 507 for programming the LPs is either stored internally in the programmer, or provided by an external computer via USB port 509 or other means of data transmission. Another embodiment utilizes a mechanical means to scan a linear array of light emitting elements over the LP's to program them.

Figs. 6A-6D - Waveforms and timing

Fig. 6A - Run mode:

[0055] The LP's are driven by the square wave signals from the driver 602, 603 at a constant rate. The LP's advance frame count on every edge, thusly, the speed of playback can be varied by altering the frequency of the drive signals. The normal speed of playback is 30 frames per second, or one frame every 33.3 ms, 600 making the nominal drive frequency 15 Hz. Every N frames (end of the video segment) the driver produces a RESET pulse 601 that is substantially shorter in time, which resets the frame pointer back to the first frame. The next edge is ignored, allowing the timing of frame display to resume normally on the next subsequent edge.

Fig. 6B - Command Mode:

[0056] The command mode starts with both driver outputs LOW 604, which ensures that the LP sees no power. The driver then turns the power on 605 for a minimum time, allowing the LPs to start in a known reset. The next eight pulses are interpreted as data, a long pulse is recognized as "0" 606 and a short pulse is recognized as a "1" 607, each edge is detected so pulses can be of either polarity, only the pulse time is measured.

Fig. 6C - Read Frames Mode:

[0057] After receiving a READ FRAMES command, the LP's enter READ FRAMES mode, where all LP's report the number of video frames programmed into them collectively back to the driver. The LPs transmit frame count data in binary format by turning the LEDs on or off, substantially changing the current level flowing in the LP circuit. The change in current level is detected by monitoring the signal from the current sensor 203; a logical “1” is registered by detecting the LED current pulse 608, which is delayed from the main edge for easy detection, a logical “0”, is detected by the absence of this pulse 609.

Fig. 6D - Programming Mode:

[0058] In programming mode, clock signals are delivered electrically via the drive signals, while the data 611 is delivered optically. The optical CLOCK SYNC 610 is either generated by the programmer or derived from the programming video sync field.

Video Programming Raster- Fig. 7

[0059] The raster area of the video projector 702 covers the entirety of the sign, so each LP can be programmed by the projector light beam. A small area is reserved for the sync pulse 701, which clocks the driver signals and LPs during programming. The X,Y Coordinates of each LP 703 are used to map the pixels in the video raster to match up with physical position of the LPs on the sign 700. The programming beam is generated by a multiple pixels 704 covering an area large enough to ensure that the light sensor in the LP adequately receives the optical signal.

Mechanical Configuration Fig.8

[0060] This illustration shows four LPs mounted on a rigid mounting substrate material comprising two conductive layers 800, 801 laminated to a central electrically insulating core material 803. The LPs are mounted in holes drilled through the material 812 and fixed in position with screws 802. The metallic housing of each LP 804 makes electrical contact with the top conductive layer 800 while the screw and washer 802, 810 make electrical contact with the rearmost conductive layer 801, the threaded body of the screw engages with the center electrode pin 809, which is insulated from the housing by plastic bushing 805, completing the electrical circuit path to the LP. A small circular Printed Circuit Board 813 mounts the electronic components: RGB LED 806, light sensor 807, and MCU 808.

Fig.9 START MODE FLOWCHART

[0061] When power is first applied to the LP, a timer 900 is started, while the polarity detector input is polled for input state changes 903, if the timer 900 elapses without detecting a change of input state the LP enters SELF-TEST MODE 902. If the input state changes within the 1-second time window, the LP enters COMMAND mode 904, and a second pulse timer 905 is reset and started. Subsequent edge transitions are detected 906, if more than 19 msec. has elapsed since the last edge 907, then the command bit is considered to be equal to logical 1, and a 1 is rotated into the COMMAND REGISTER 908. If the time period is less than 19 msec., the bit is considered to be logical 0, and a 0 is rotated into the COMMAND REGISTER 910. This process is repeated until all 8 bits of the command are received 911, then the value in the COMMAND REGISTER is compared against values for PROGRAM mode 913 and SEND FRAMES mode 914, if the command does not match, the program defaults to RUN MODE 915.

Fig.10 SELF-TEST MODE FLOWCHART

[0062] This mode is used for test purposes to validate the function of all internal systems of the LP and to identify the current firmware version of programmed into the LP.

[0063] The first portion of SELF-TEST mode conveys the firmware version as an 8 bit binary number (1000) by flashing the RED light source momentarily for logic ‘1’ and GREEN for logical ‘0’, each bit is separated by a momentary flash of BLUE, which signifies an inter-digital space.

[0064] Once the firmware ID is completed, the LP enters a test mode loop that tests all relevant features of the LP, the logical state of the input is detected 1002 and the RED 1003 or GREEN 1004 light is illuminated depending on the detected polarity, this mechanism allows testing of the POLARITY DETECTOR circuit (Fig. 1,103) for correct function. The light is subsequently turned off after a delay 1005, which is proportional to the illumination level impinging on the light sensing element (Fig.1 106) in such a way that the light flashes on and off at a rate proportional to the light level, validating the function and relative performance of the light sensor and associated systems.

Fig.11 PROGRAM MODE FLOWCHART

[0065] Program mode receives optical data signals from the light sensor (Fig. 1,106) and electrical clock signals from input polarity reversal, via polarity detector (Fig.1, 103). The two signals CLOCK and DATA are used to clock data serially into the LP for storage in the LP's onboard memory unit (Fig. 1, 105).

[0066] The first eight clock pulses received by the LP consist of eight alternating light / dark signal levels 1101 that are averaged and measured to calculate the threshold for determining the logical value of subsequent light pulses 1102. If the phase of the pulses is incorrect, the process fails 1103, and the RED LED is illuminated to indicate the failure 1104. If the phase of the calibration pulses are correct, the signal level is then checked to determine if it meets minimum level requirements, if this test fails 1105, then the LED is illuminated YELLOW 1106, indicating failure. If the signal level is good, the LP then receives the number of frames to be programmed as a 16 bit binary number 1107, coded in the same manner as a commands, as shown in Fig. 6B.

[0067] The number of frames received data is tested for a zero value 1108, and for a value greater than the available memory 1110, failures illuminate the LED with a PURPLE color 1109, 1111, if the tests fail.

[0068] If the number of frames received is valid, the LP begins receiving 'frames' of 24 bit RGB data, until the number of data frames is equal to the initial value received 1112, subsequently, the LP receives an 8 bit checksum that is calculated by exclusive OR of all the data bytes together, if the checksum data does not match the internally calculated value 1114, the BLUE LED is illuminated 1115, indicating CHECKSUM FAIL. If the checksum is correct, the GREEN LED is illuminated 1117, indicating a successful programming procedure.

Fig.12 RUN MODE FLOWCHART

[0069] RUN mode relies on pulse timer 1200 to determine if the incoming pulse is a normal long frame advance pulse, or a short FRAME RESET pulse. Pulse timer 1200 is reset after every edge transition to ensure that the pulse width can be measured at the next edge. Every edge transition is detected by polling the input 1201, this event triggers evaluation of the timer 1202, if the pulse is of normal length, longer than 19 msec., the frame pointer 1204 is advanced to point to the next frame, and the frame count is then tested against the end of memory 1206, and reset if the count equals the end of memory value, 1207. The new frame pointer value is then used to fetch the next RGB value from memory 1208, and the LED output is updated with the new frame value, 1209.

Fig.13 SEND FRAMES MODE FLOWCHART

[0070] SEND FRAMES mode utilizes the LED system in the LP to vary the current flow in the drive circuit to communicate the number of frames stored in the LPs. Upon entering SEND FRAMES mode, the number of frames in memory is fetched 1300, and the working shift register is loaded with the value 1301. When an input edge transition is detected 1303, the shift register is rotated 1305 to determine the bit value to be sent, 1 or 0, 1306. A delay 1304 is then started to place the current pulse in the middle of the input pulse clock period, (fig. 6C - 608) avoiding coincidence with current spikes normally caused by the input polarity switching. If the value of the bit rotated out of the register is = ‘1’, the LED is turned on to create a large current flow 1308. The LED is turned off at the next input edge and the process is repeated until all 16 data bits have been clocked out serially. The program automatically enters RUN MODE 1310 after the frame data is sent.

Sign Design and Fabrication Process

Fig. 14A, 14B

[0071] Several categories of software tools need to be used at various stages of the sign creation process. Developing and refining this toolchain is essential to the value proposition of the LP system. This toolchain may be a combination of off-the-shelf software, custom software made with “authoring tools”, or software coded more-or-less from scratch. The proportion of these types of software (and their exact feature set and specification) will change and adapt with time but every solution strives to satisfy specific design criteria and address specific phases of the creation process. These phases can most broadly be split into two categories: the first uses software that determines or implements LED Pixel placement and the second software tools for authoring lighting effects on the pixels. A third, parallel category of tools relates to giving customers the ability to accurately pre-visualize a potential sign design and the lighting effects programmed onto it and, furthermore, to customize this sign and content in an easy and intuitive way.

Exemplary Design Criteria for Software

1. Allows for precision in design and fabrication to industry standards
2. Automates as much of the process as possible, reducing labor cost
3. Simplifies and streamlines the process as much as possible, reducing the training and technical expertise required to use the software successfully
4. Uses the latest cloud and web-based technologies to facilitate logistics, customer tracking, new fabrication facility setup and other considerations.

Toolchain Phases

1. LED Pin Positioning Phase

Overview

[0072] In the design phase, graphic design collateral material from the customer (or at least a design concept) can be translated into an actual sign concept, set to the appropriate scale and with specific design and material specifications. The distinctive part of this process for the LP system is software that takes advantage of the fact that the LPs can be easily spatially distributed anywhere across the sign, without needing to conform to an underlying grid or specific layout pitch. This gives tremendous design freedom: the ability to easily implement many “non-regular” LED placement layouts. However, this ability also raises a challenge to create software solutions that help clients create and choose between these unique layouts.

[0073] Another important capability is a way to test the basic design against different quantities of LED pins (laid out in potentially different, but quantity-specific optimum ways) so that the client can easily experiment with different solutions across a matrix of price points and design complexity.

Software

[0074] Various software solutions can be used in accordance with the invention for the LED Pin Positioning phase. For example, in a presently preferred approach, 2D vector graphic design software can be used, such as, Adobe Illustrator or Corel Draw or 2D or 3D CAD software like AutoCAD, Vector Works, Rhino, Solidworks etc. The former category is generally the native format of the client collateral material, while the latter is generally the format a traditional sign fabricator would use, particularly to pass data to CAM equipment to actually fabricate the sign. Other options can include utilization of some proprietary LED sign software that could be modified to add the LP requirements, or software could be written that incorporates the important elements of the workflow of using both classic graphic design and CAD software.

[0075] Features: In a detailed aspect of an exemplary embodiment, the first layout feature that any LP solution (or combination of solutions) needs to solve is the ability to distribute the LPs evenly along a path, and where the overall density can be specified. Adobe Illustrator does this with the Blend Tool and CAD software does this with the "MEASURE" and "DIVIDE" commands. The next feature set involves creating the "fills" for shapes: the ability to place LPs within a bounded shape where the placement of the pins is at a specifiable density and the LPs appear to be more-or-less evenly distributed. Because LPs do not need to be arranged in grids, the software should take maximum advantage of this capability to create interesting non-grid layouts. One technique (available in both Illustrator and CAD software) is to offset the path defining the shape inward to create multiple "concentric" paths into the center. The new offset paths are then also processed to also distribute pins along these respective paths in the same manner as the first feature above. If the offset amount matches the spacing of pins along each path, then it is often possible to use this technique to get an optimal, non-grid, evenly spaced LP placements filling a given shape. This technique has the added advantage of allowing the software used in the later Content Design Phase to create animation effects that reference the "groups" of **LPs** that belong to the respective paths to animate them separately, creating interesting effects. In any case, the software should also allow the user to edit the position of LPs and to manually add or delete LPs at will. The software should provide data feedback on the number of LPs used (for cost control) and a full set of measuring and general object distribution tools.

[0076] Resulting Output: In this exemplary approach, the LED Pin Positioning phase provides **two** files that are utilized for other phases in the overall process, as follows:

1. A precisely-scaled CAD file that provides all the information necessary to fabricate the sign shape - including CAM-ready instructions to cut all the physical materials used in the sign, including the layer of conductive material that will power the LPs – as well as to place the LPs accurately into position.
2. A scaled version of the LPs positioning for use in the Content Design Phase. This could be as simple as a scaled graphic showing the placements or even a data file listing them.

2. Content Design Phase

Overview

[0077] The content design phase provides the lighting effects that will play on the LP system. In an exemplary embodiment, one of the distinctive features of the LP system is that these lighting effects are uploaded to flash memory embedded in each of the LPs individually and they then play back in synchronization when deployed on the sign. Many various authoring strategies for generating content for the LP system can be used without departing from the invention. For purposes of illustration, two exemplary authoring strategies are discussed, as follows:

1. <u>Sampled Animation</u>. With this technique animation effects are authored in a standard "digital video" matrix that is scaled to cover all the LP positions. Generally, this video is authored at a much higher resolution then the general LP resolution – so that each LP would sit "over" many pixels of the "underlying" digital video. For each frame of this animation, the software will sample the area under each LP position and average the color within that sample to assign an RGB value for each pin for that frame. One example would be a soft-edged stripe that moves across the frame, or a shifting linear or radial color gradient. The software would sample these animations at the LP positions and then, after the sampled data is encoded into the LPs (in the Encoding

Phase described next), an approximation of this animation will play back "accurately spatialized" across the LPs in the final sign deployment.

2. Direct LP Animation. The LPs are assigned colors and effects directly and specifically. This approach is best explained by example. Suppose you want to create some lighting effects for a sign, you are looking at the LP layout, and you then proceed by selecting specific LPs or groups of LPs to light in specific sequences, with specific colors and with specific effects. One example is a "chase", where LED pixels defining the perimeter path of a shape light up in sequence. Another example would be where LED pixel groups defined using the offset path technique described in the previous section all trigger in sequence, creating a "concentric animation" effect. Another example involves creating animation effects where each pixel animates independently using a specific algorithm. For instance, each pixel could interpolate back and forth between two colors. By randomizing the start position of each interpolation, each pixel could oscillate between those two colors in a different phase, creating a pleasing visual effect.

Software

[0078] Various software solutions can be used in accordance with the invention for the Content Design phase. For example, in a presently preferred approach, for the "Sampled Animation" technique, digital video effects software like Adobe After Effects is effective in creating the "background" digital video patterns and animations. The scaled graphic version of the LP positons created in the LP Positioning Phase can be imported to serve both as a guide to structuring the animation and as a pre-visualization of the effect once it plays back from just the LP locations (when the graphic is used as a mask to show only the parts of the animation under the pins). To perform the actual sampling we use Cycling '74's MAX/MSP/Jitter development environment to write a "patcher" that uses the LP positon data to sample the movie created within this environment or imported from After Effects. For the "Direct LP Animation" technique, we use the Adobe Flash authoring environment, which allows us to define each LED pixel as a "sprite object" that can be specifically and independently animated.

[0079] Other software options for the Content Design phase can include a single "integrated" software tool that combines the best features of the three software tools above would be the ideal solution. Using Adobe After Effects is particularly problematic in that its use implies an operator trained in that software. MAX/MSP/Jitter and Adobe Flash, on the other hand, are **authoring** environments where a very specific tool can be designed with a User Interface that clearly guides the content designer on how to author content.

[0080] Features: In certain embodiments, a feature of a solution would be the ability to import the LP layout made for a specific sign during the LP Positioning Phase. Another feature could be the ability to specify groups of LEDs, so that animations can be assigned to different elements of a sign separately. For instance, different letter shapes could be animated one at a time, or a group of LEDs defining the perimeter of a shape could be animated differently from the fill of that shape. The easiest to use version of the software (one version would probably not satisfy all use cases) would allow a client to simply assign and **pre**-visualize "canned" animations to the LED pixel groups. Finally, the software needs to ensure that the animation data is prepared in a format understandable by the software component accomplishing the goal of the next phase, the Encoding Phase

[0081] Resulting Output: One or more files ready for encoding that could comprise images, text, and/or video, or combinations thereof.

3. Encoding Phase

Overview

[0082] The Encoding phase is generally directed at taking the lighting animation data created in the Content Design phase and uploading it to the embedded flash memory of each LP. In an exemplary embodiment, the LP process has a unique way to do this: each **LP** has a light sensor that reads each RGB value for each frame of the animation as a series of on-or-off light flashes that represent that RGB value as a binary number. Various approaches can be used. For example, two set ups are discussed, as follows:

1. In-place, “spatialized” programming. With this technique, the sign is fully assembled and then programmed with a digital projector. All LPs are programmed in parallel, one frame at a time. The current system flashes a discrete white or black square over each LP every 60th of a second – it takes 24 pulses to program each frame. The projector has to be calibrated (by adjusting the projection distance, zoom and/or keystone correction) so that the scale of the encoded movie is correctly sized when projected onto the sign.
2. Gang-programming. With this technique, the LPs are packed into a tight grid and programmed in a specially constructed programming unit that programs then in the same way as the In-place system. In this case, the software needs to separately track which LP in the packed grid maps to which LP in the actual sign. This system is faster, requires less space and calibration time and effort, but has the disadvantage of the additional need to manage the position data associated with each LP in the non-spatialized grid. The “In-place” system has the additional benefit of allowing the sign to be re-programmed with new or edited content by putting the sign in front of the projector again and running a new encoding movie.

Software

[0083] Various software solutions can be used in accordance with the invention for the Content Design phase. For example, in a presently preferred approach the encoding is done by a Max/MSP/Jitter patcher that uses the movie-scaled position points of the LPs to read the RGB pixel data of the Content Design Phase movie at each point and then creates a 24-frame sequence of white and black boxes surrounding that point that represent the 24-bit RGB value being encoded. With the “In-place” technique this is saved as another digital movie file that has 24 frames for every frame of the source Content Design Phase movie (plus some additional frames before and after this section consisting of a beginning initialization sequence and ending with a checksum sequence, see below). With the “Gang-programming” technique the MAX/MSP/Jitter patch writes the data into a formatted text file where each line of the file (again, not including the initialization data and the closing checksum data) is a sequence of ones and zeros designating the current one bit value being sent to each LP in parallel.

[0084] Other software options for the Content Design phase can include a tool that could be combined with the Content Design Phase tool in written-from-scratch proprietary version of the software. Other versions of this software might encode LED pin color data in a **format** different than 24 bit RGB data to save transmission time, for instance by using 16 bit data or a CLUT (Color Lookup Table) using 8 bit or less values.

[0085] Features: In an exemplary embodiment, regardless of the techniques used, the encoding preferably follows a certain structure. One part of the data sent, **discrete** from the RGB bit data transmitting the current bit value for each LED pixel being programmed, is a clock pulse that toggles on and off for each bit of the transmission so that the data across LPs is clocked in sync. As previously mentioned the data also has a “preamble” consisting of an initialization sequence. This consists of 8 alternating pulses of black and white sent to all the LPs (they all get the same data) that is the Calibration Sequence, followed by 16 flashes (again, all the LPs get the same data) that represents in binary the number of frames to be programmed, so that each LP knows how many frames of encoded light pulses it needs to read into memory. Then comes the data for each LP (here each LP gets only its own data). Finally, a set of eight more pulses is sent consisting of a checksum value for each LP to confirm that there were no errors in the transmission.

[0086] Resulting Output: In the case of the “In-place” technique, the finished output of this phase can be a digital video file containing **the** encoded values for the LP programming sequence, where each frame of this movie transmits one bit of data simultaneously to all the LPs. In the case of the “Gang-programming” technique, the finished output of this phase can be a text file that contains the same data as the previous technique, except written as sequences of ones and zeros rather than represented as black and white boxes laid out in a spatial representation of the LP placements.

[0087] The present invention has been described above in terms of exemplary embodiments so that an understanding of the present invention can be conveyed. However, there are other embodiments not specifically described herein for which the present invention is applicable.

Therefore, the present invention should not to be seen as limited to the forms discussed, which is to be considered illustrative rather than restrictive.

[0088] Although the invention has been disclosed in detail with reference only to the exemplary embodiments, those skilled in the art will appreciate that various other embodiments can be provided without departing from the scope of the invention, to include any and all combination of features discussed herein.

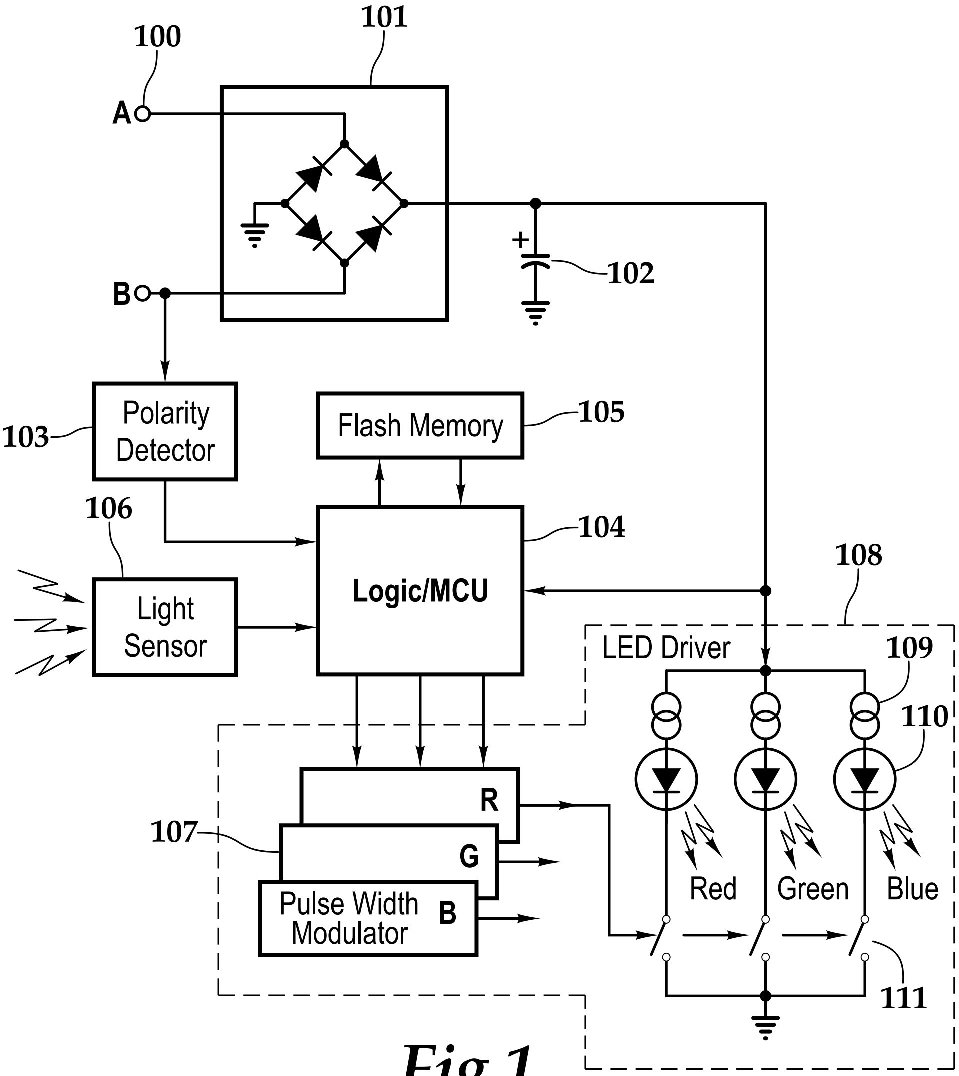
100
101
A
B
102
+
103
Polarity Detector
Flash Memory
105
106
104
Logic/MCU
Light Sensor
108
LED Driver
109
110
107
R
G
Pulse Width Modulator
B
Red
Green
Blue
111

Fig.1

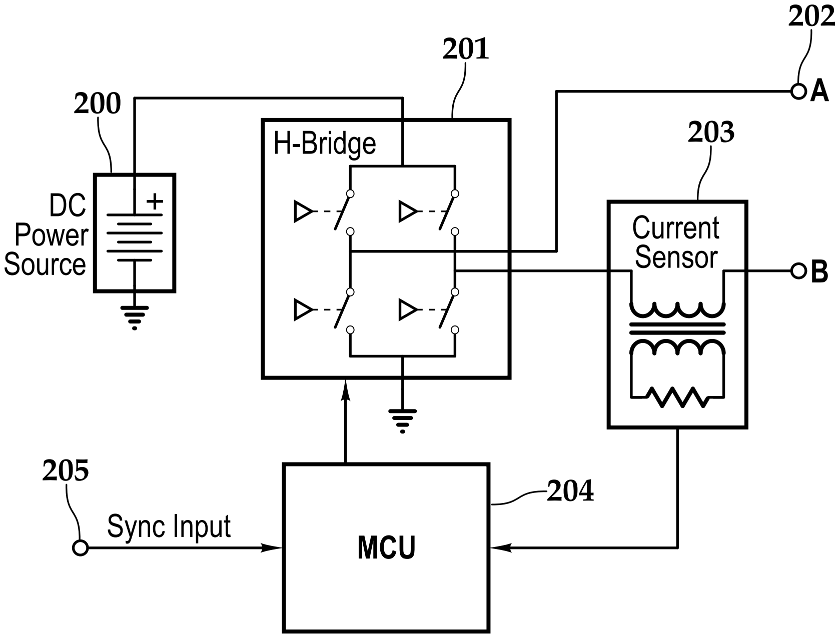
202
201
200
A
203
H-Bridge
DC Power Source
+
Current Sensor
B
205
204
Sync Input
MCU

Fig.2

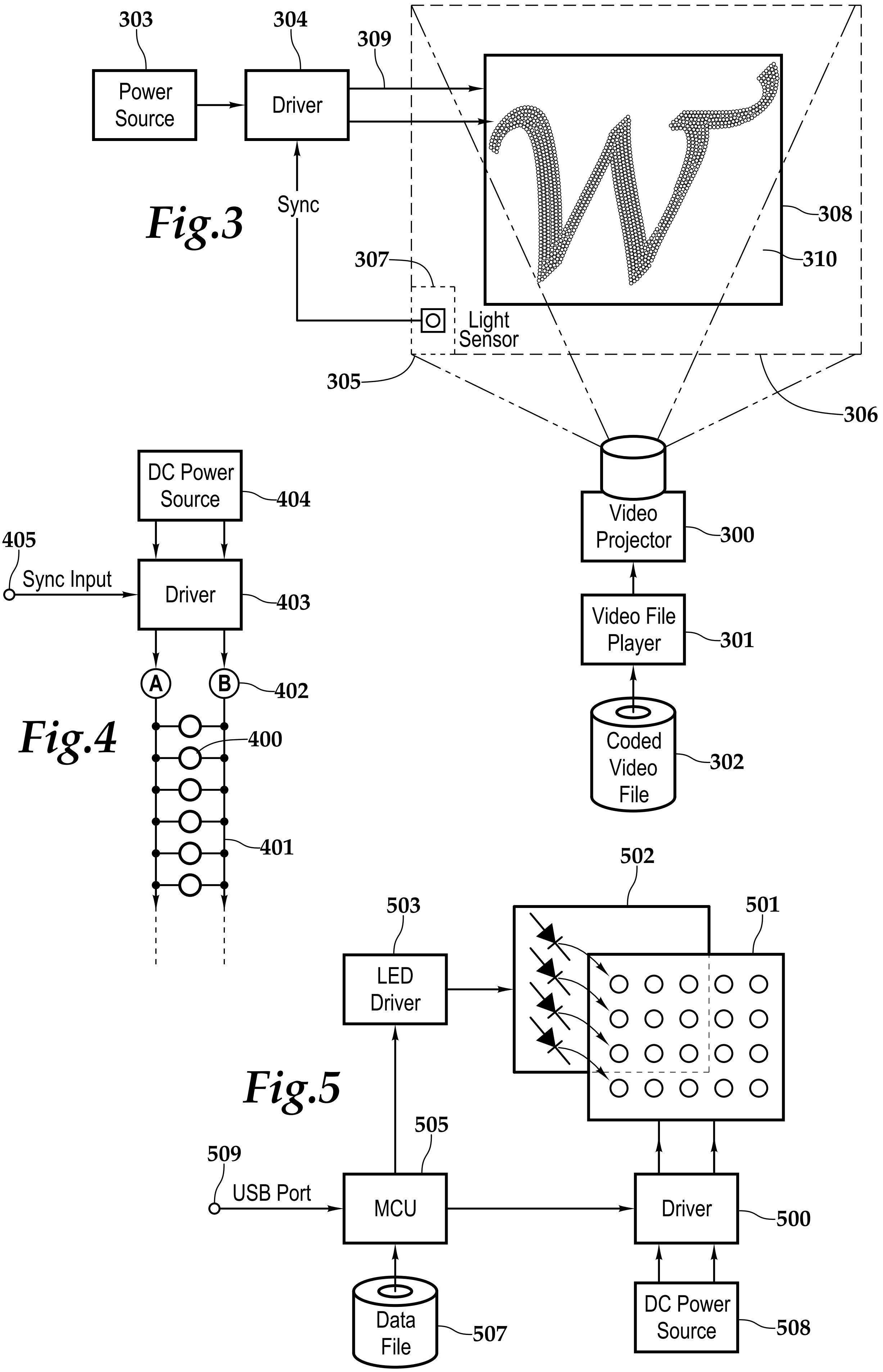

Fig.3

Fig.4

Fig.5

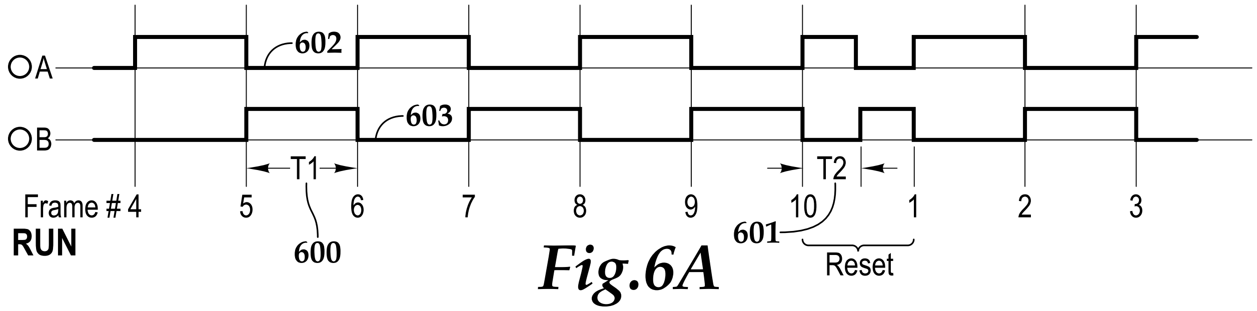

Fig.6A

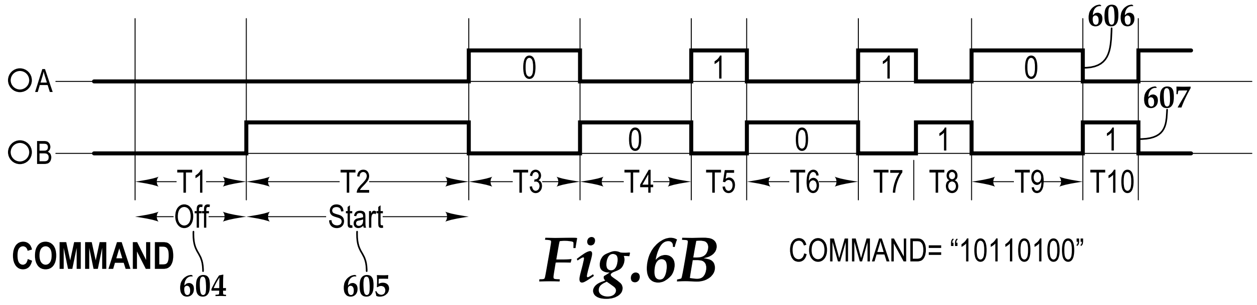

Fig.6B

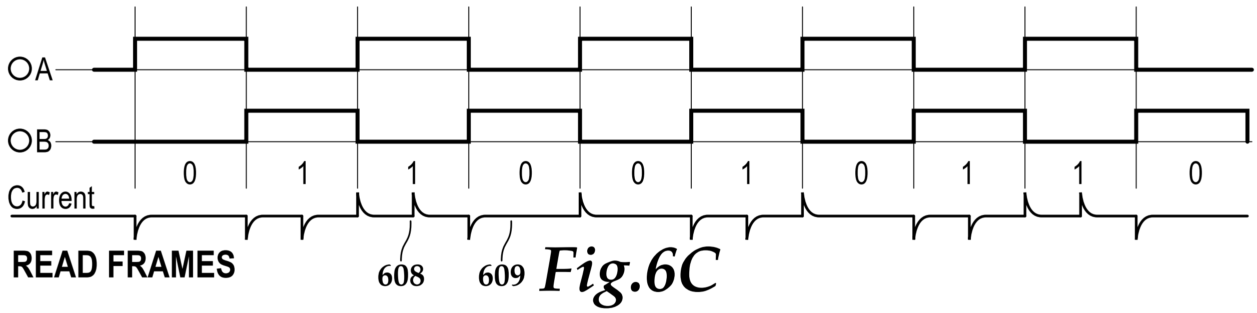

Fig.6C

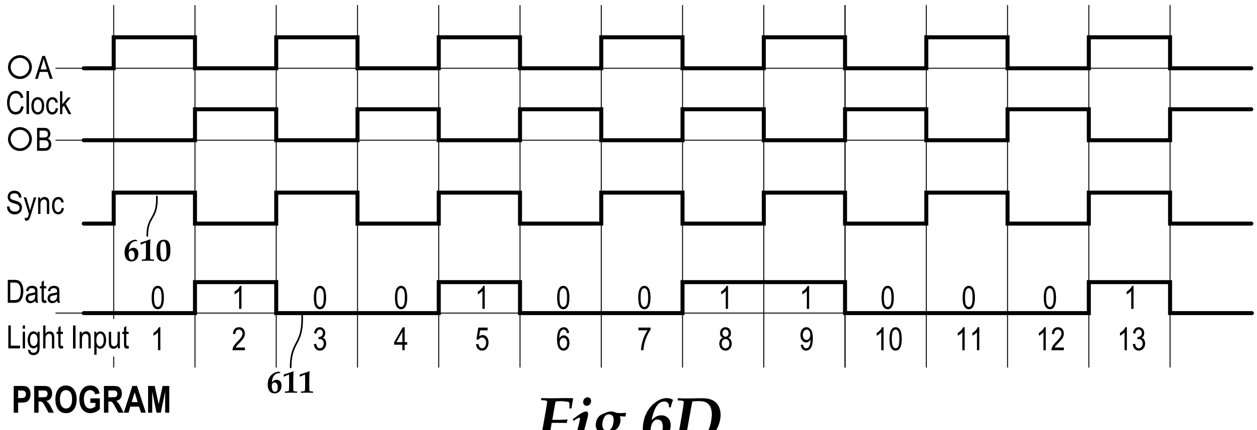

Fig.6D

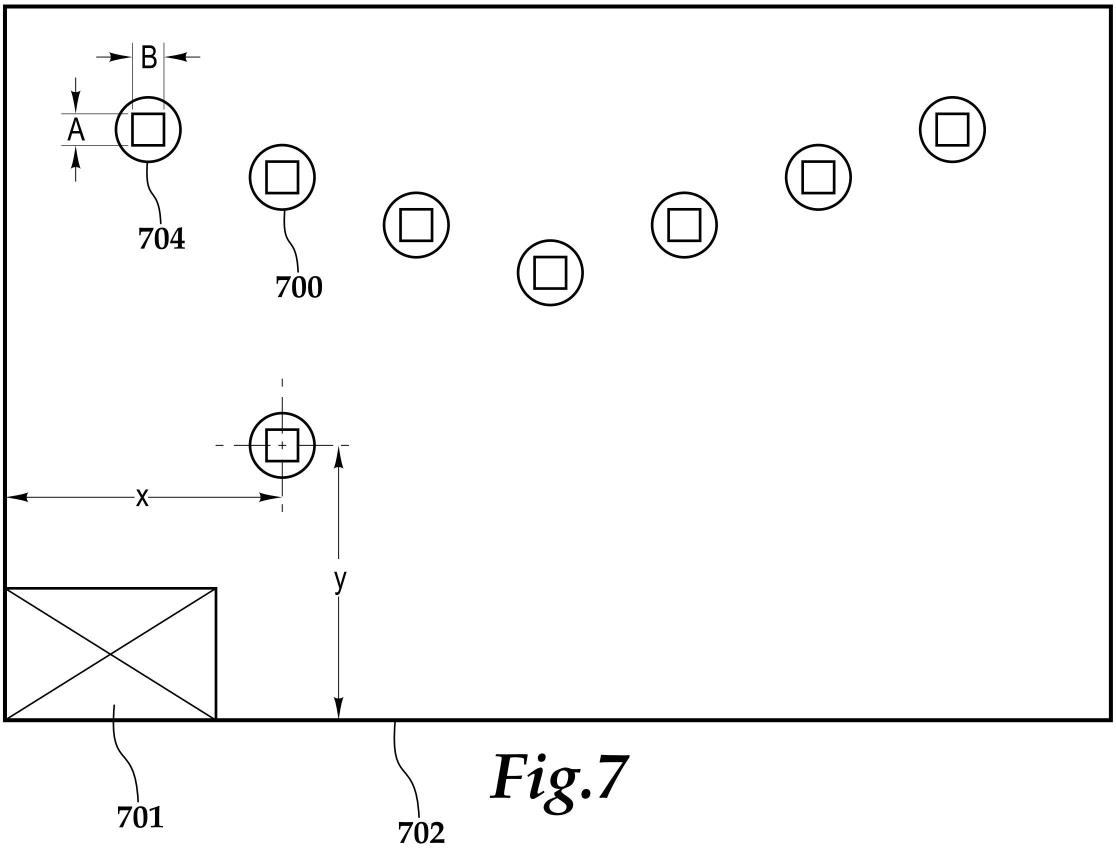
B
A
704
700
x
y
701
702

Fig.7

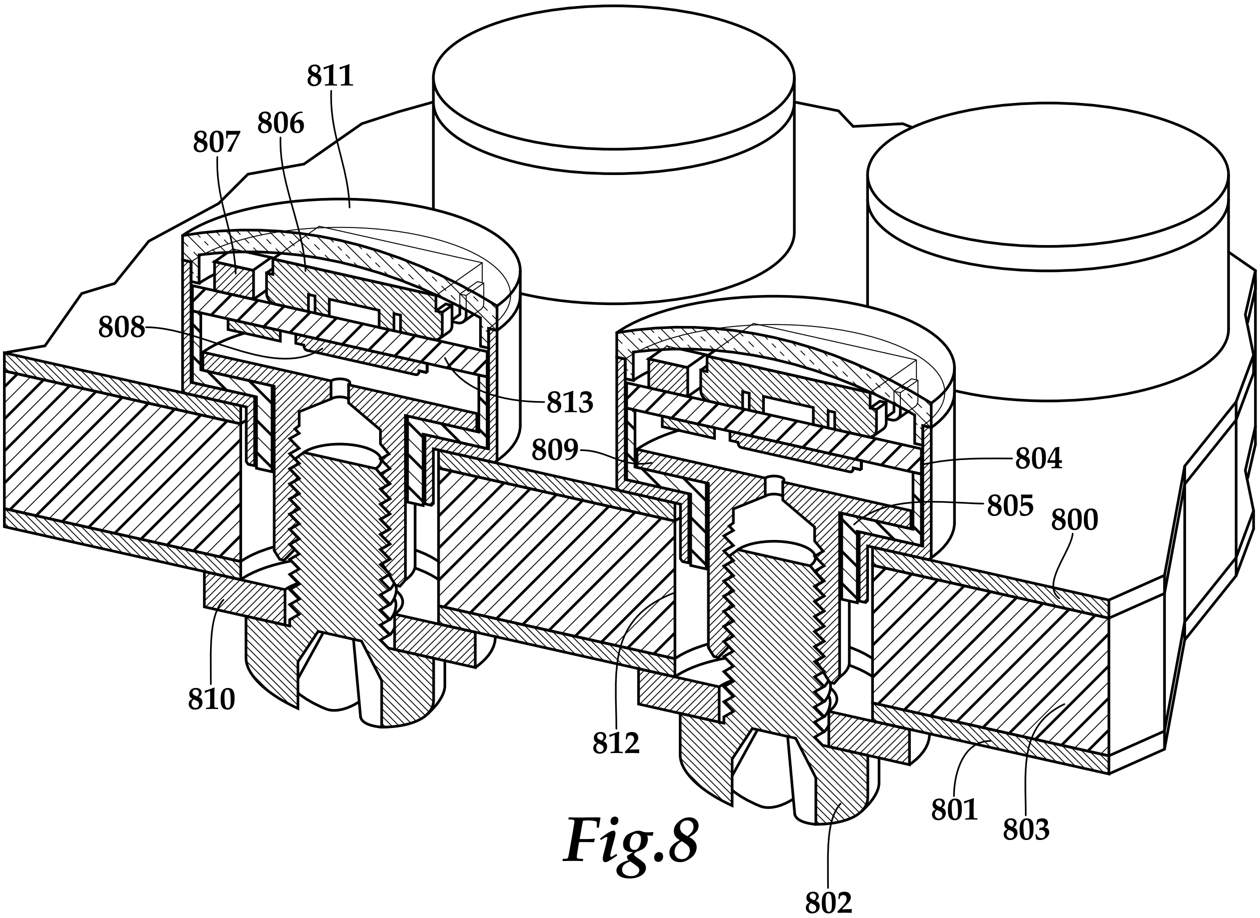
811
807
806
808
813
809
804
805
800
810
812
801
803
802

Fig.8

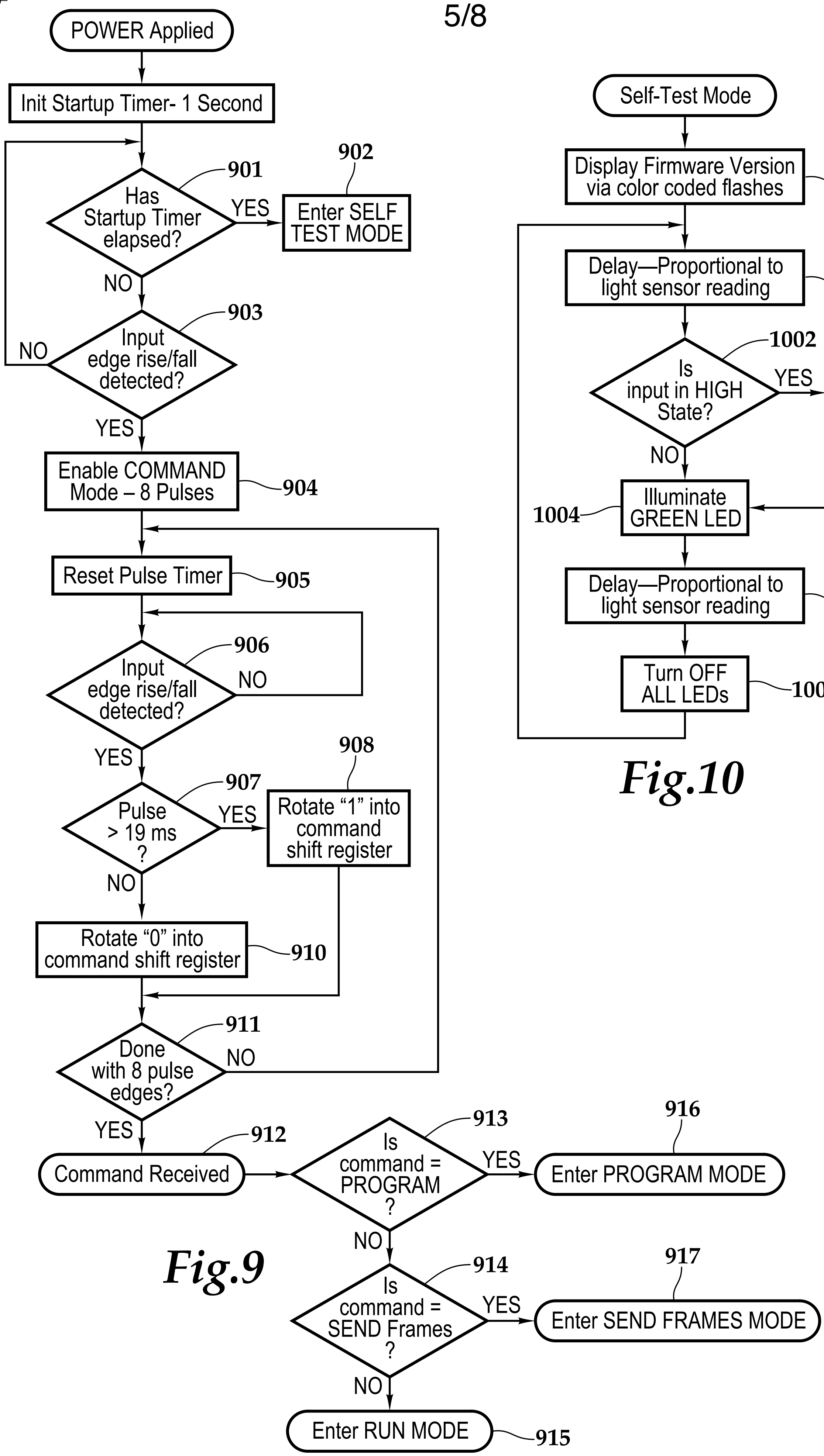
POWER Applied
Init Startup Timer- 1 Second
Has Startup Timer elapsed?
901
YES
902
Enter SELF TEST MODE
NO
Input edge rise/fall detected?
903
NO
YES
Enable COMMAND Mode – 8 Pulses
904
Reset Pulse Timer
905
Input edge rise/fall detected?
906
NO
YES
Pulse > 19 ms ?
907
YES
908
Rotate "1" into command shift register
NO
Rotate "0" into command shift register
910
Done with 8 pulse edges?
911
NO
YES
Command Received
912
Is command = PROGRAM ?
913
YES
916
Enter PROGRAM MODE
NO
Is command = SEND Frames ?
914
YES
917
Enter SEND FRAMES MODE
NO
Enter RUN MODE
915

Fig.9

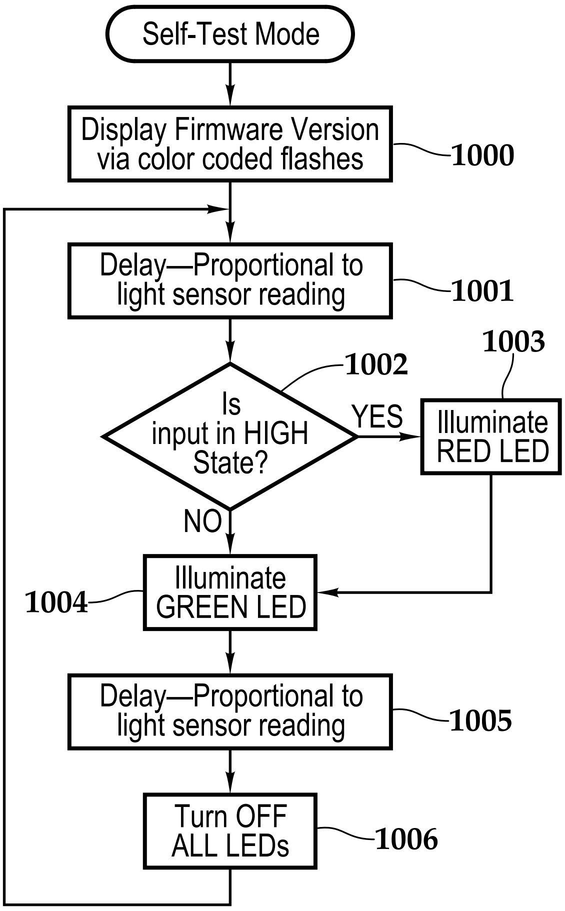
Self-Test Mode
Display Firmware Version via color coded flashes
1000
Delay—Proportional to light sensor reading
1001
Is input in HIGH State?
1002
YES
1003
Illuminate RED LED
NO
1004
Illuminate GREEN LED
Delay—Proportional to light sensor reading
1005
Turn OFF ALL LEDs
1006

Fig.10

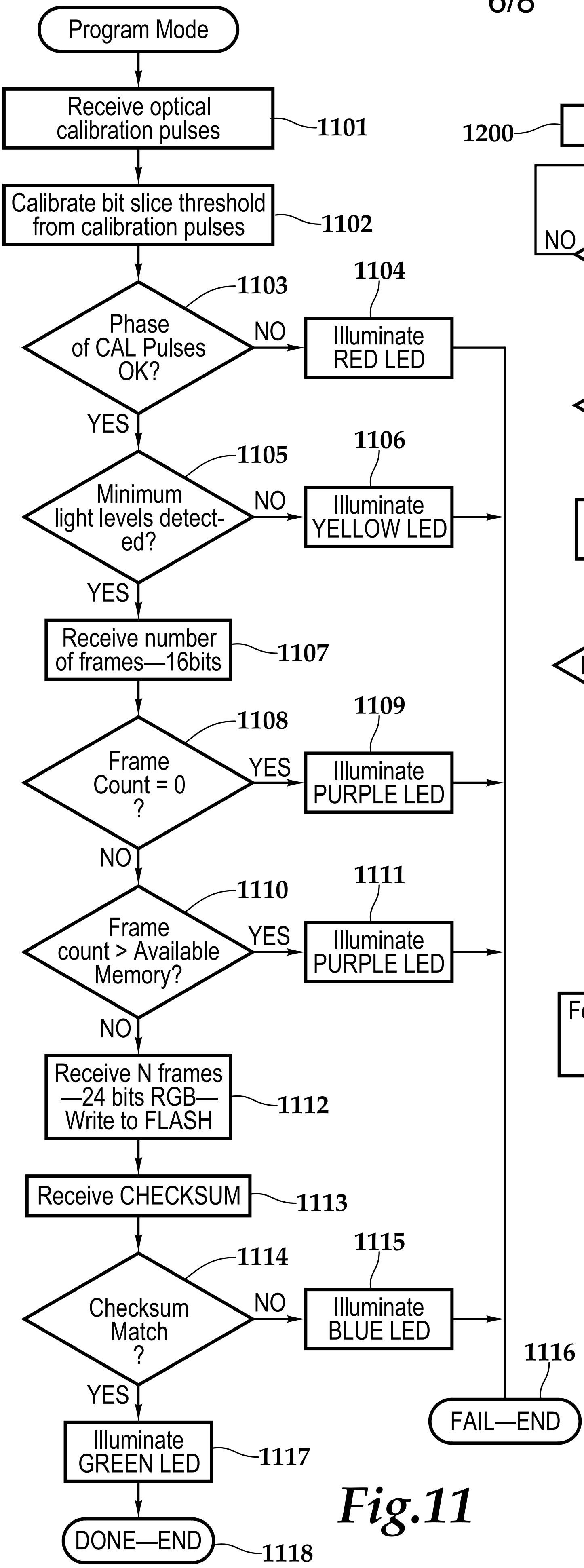
Program Mode
Receive optical calibration pulses
1101
Calibrate bit slice threshold from calibration pulses
1102
1103
Phase of CAL Pulses OK?
NO
1104
Illuminate RED LED
YES
1105
Minimum light levels detect-ed?
NO
1106
Illuminate YELLOW LED
YES
Receive number of frames—16bits
1107
1108
Frame Count = 0 ?
YES
1109
Illuminate PURPLE LED
NO
1110
Frame count > Available Memory?
YES
1111
Illuminate PURPLE LED
NO
Receive N frames —24 bits RGB— Write to FLASH
1112
Receive CHECKSUM
1113
1114
Checksum Match ?
NO
1115
Illuminate BLUE LED
1116
FAIL—END
YES
Illuminate GREEN LED
1117
DONE—END
1118

Fig.11

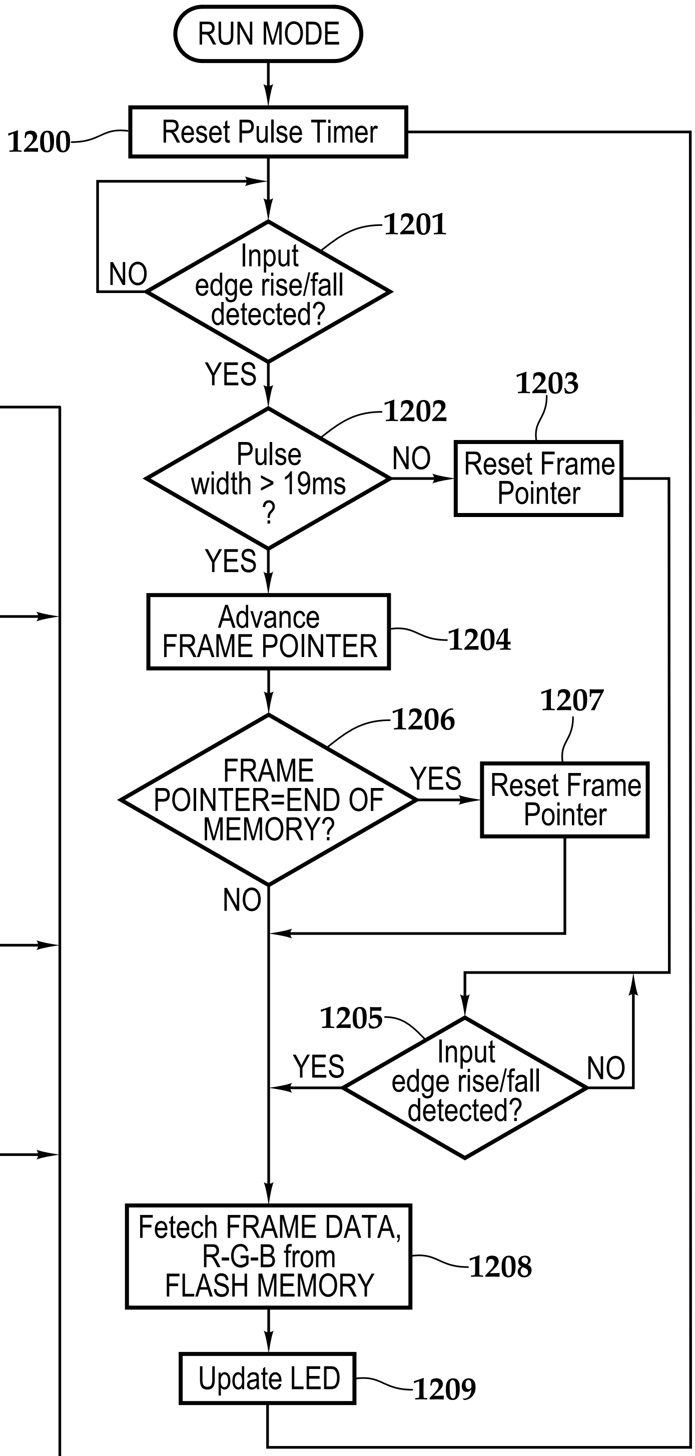
RUN MODE
1200
Reset Pulse Timer
1201
NO
Input edge rise/fall detected?
YES
1202
Pulse width > 19ms ?
NO
1203
Reset Frame Pointer
YES
Advance FRAME POINTER
1204
1206
FRAME POINTER=END OF MEMORY?
YES
1207
Reset Frame Pointer
NO
1205
YES
Input edge rise/fall detected?
NO
Fetech FRAME DATA, R-G-B from FLASH MEMORY
1208
Update LED
1209

Fig.12

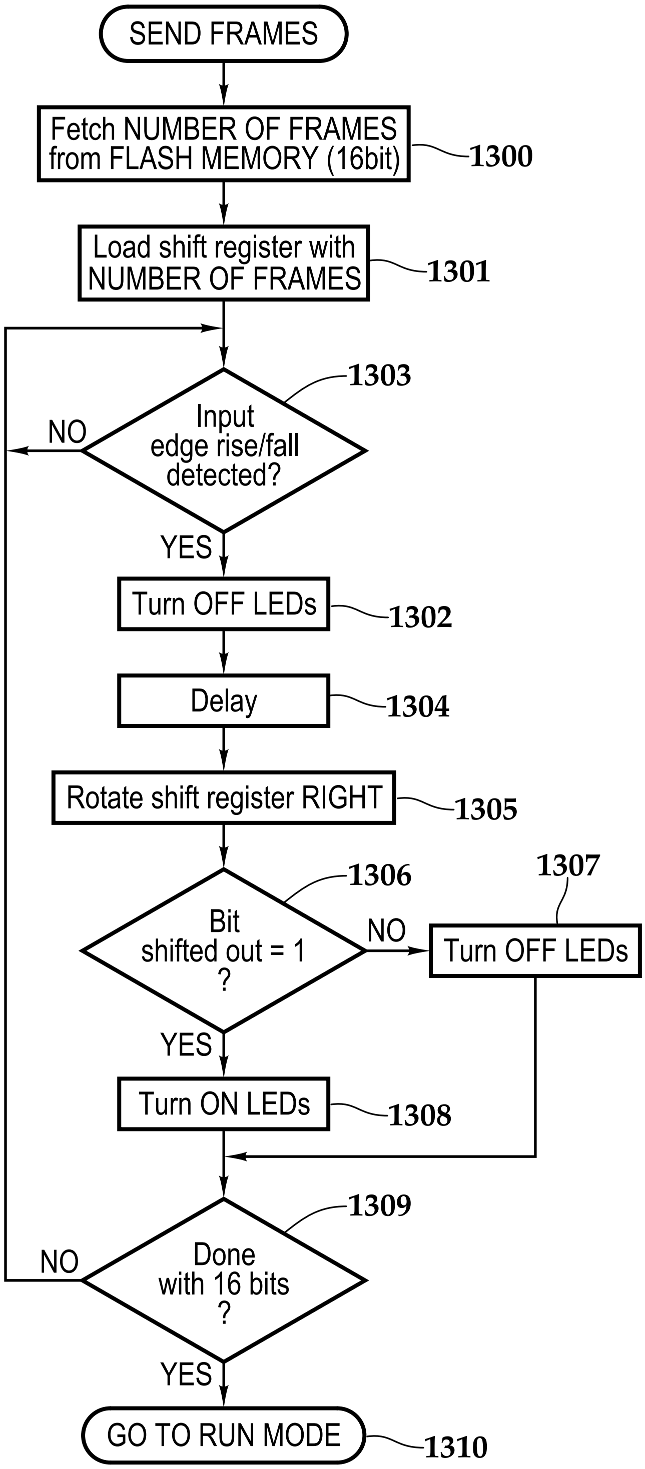
SEND FRAMES
Fetch NUMBER OF FRAMES from FLASH MEMORY (16bit)
1300
Load shift register with NUMBER OF FRAMES
1301
1303
Input edge rise/fall detected?
NO
YES
Turn OFF LEDs
1302
Delay
1304
Rotate shift register RIGHT
1305
1306
1307
Bit shifted out = 1 ?
NO
Turn OFF LEDs
YES
Turn ON LEDs
1308
1309
NO
Done with 16 bits ?
YES
GO TO RUN MODE
1310

Fig.13

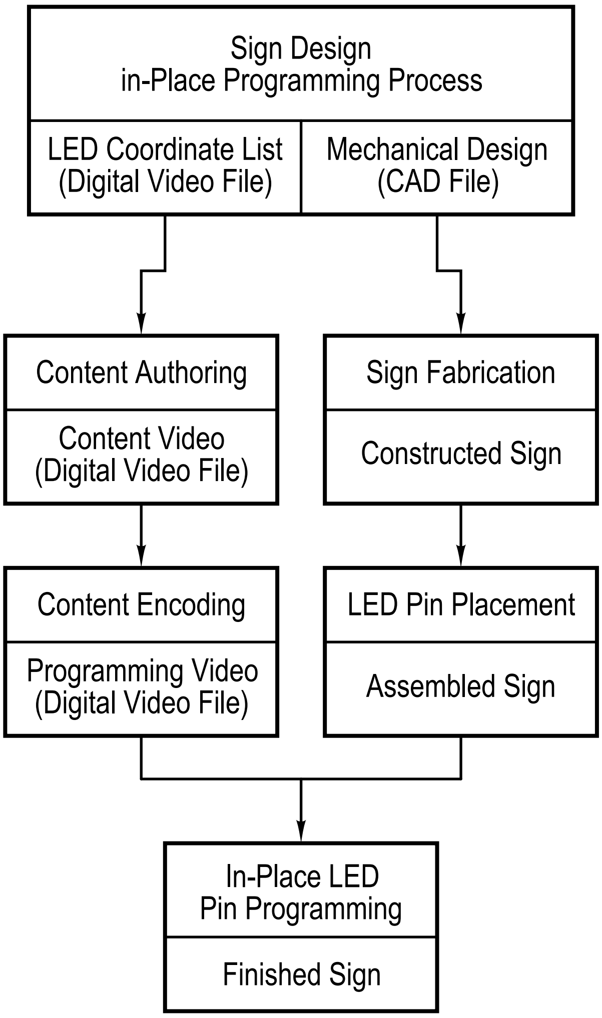

Fig.14A

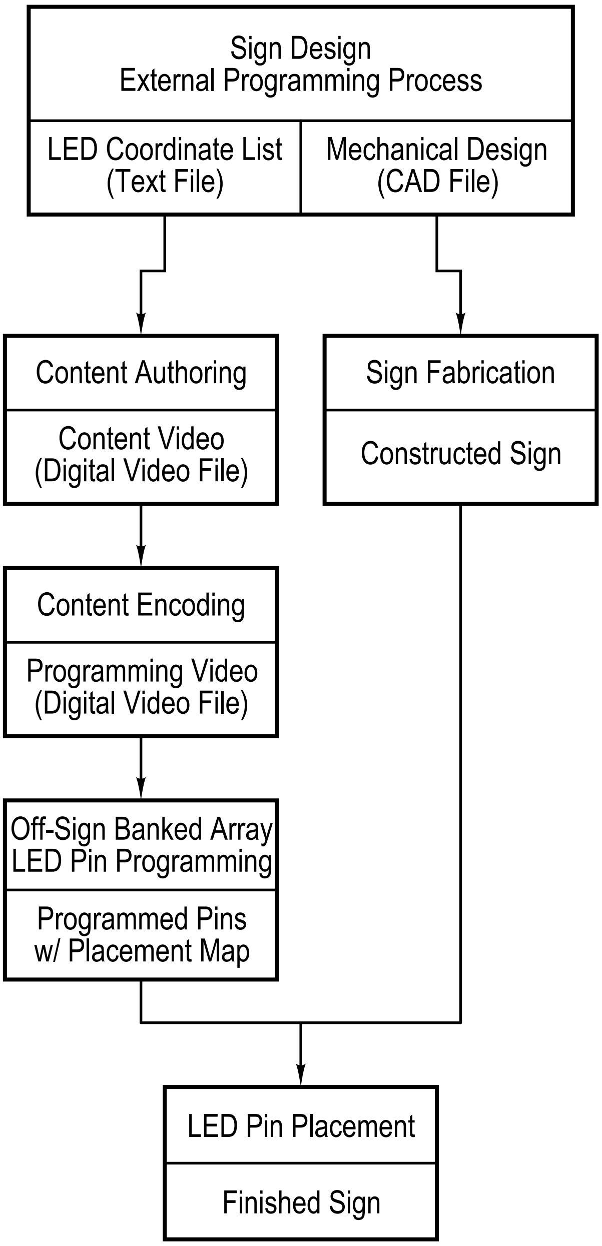

Fig.14B