US 20210138356A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2021/0138356 A1

(43) Pub. Date: May 13, 2021

LEWIS

(54) LED SCENE CONTROLLER FOR A MODEL TRAIN SYSTEM AND RELATED METHODS

(71) Applicant: **James Bevan LEWIS**, Orlando, FL (US)

(72) Inventor: **James Bevan LEWIS**, Orlando, FL (US)

(21) Appl. No.: **17/092,646**

(22) Filed: **Nov. 9, 2020**

Related U.S. Application Data

(60) Provisional application No. 62/932,918, filed on Nov. 8, 2019.

Publication Classification

(51) **Int. Cl.**

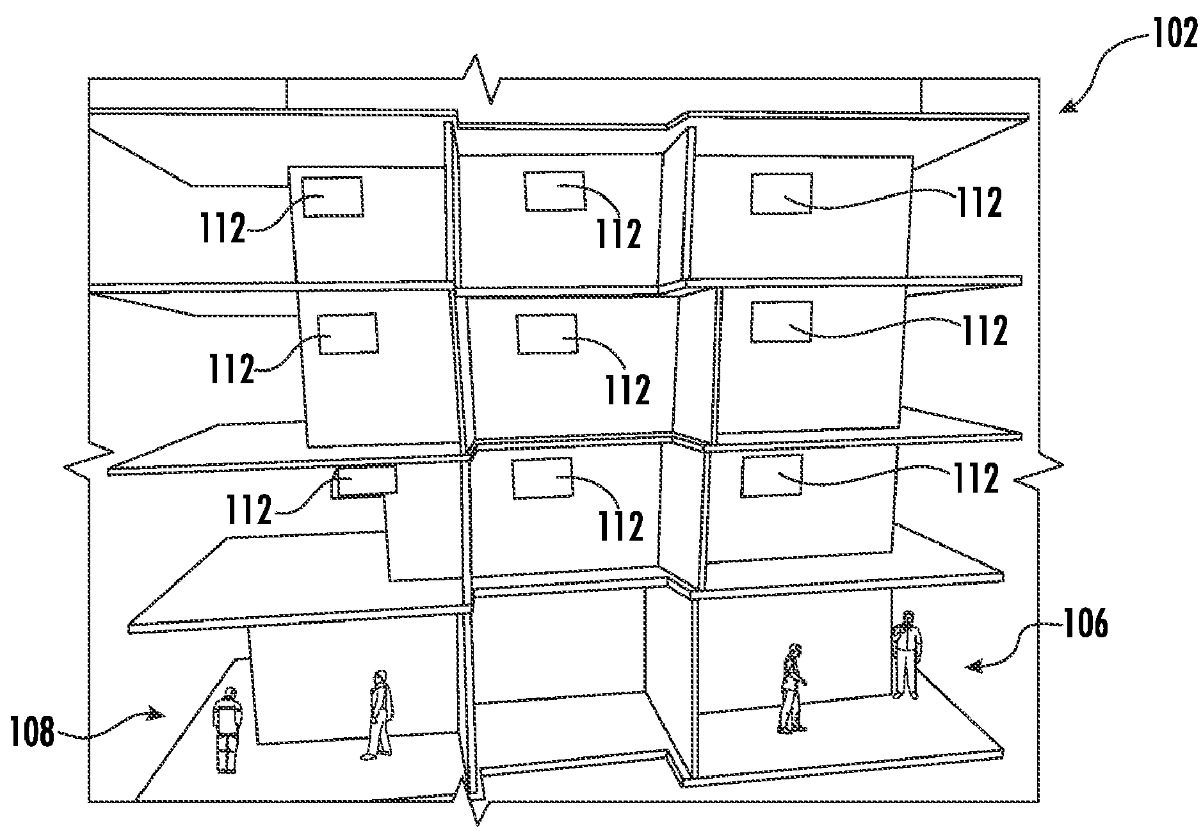
A63H 19/24	(2006.01)
A63H 19/34	(2006.01)
A63H 19/20	(2006.01)

(52) **U.S. Cl.**

CPC ***A63H 19/24*** (2013.01); *A63H 2019/246* (2013.01); ***A63H 19/20*** (2013.01); ***A63H 19/34*** (2013.01)

(57) **ABSTRACT**

An LED scene controller for a model train system includes a printed circuit board (PCB) having a decoder and a plurality of light emitting diode (LED) ports mounted thereon, and a plurality of special effects components with each coupled to a respective LED port. The decoder includes a processor and a memory coupled to the processor, where the processor is coupled to the plurality of LED ports. The decoder is configured to receive a digital command control (DCC) signal comprising a plurality of configuration variables from a DCC hand controller and to store the plurality of configuration variables in the memory. The processor is configured to read the plurality of configuration variables to control the plurality of special effects components coupled to the plurality of LED ports. The decoder is configured to automatically switch to an animation mode when no DCC signal is detected.



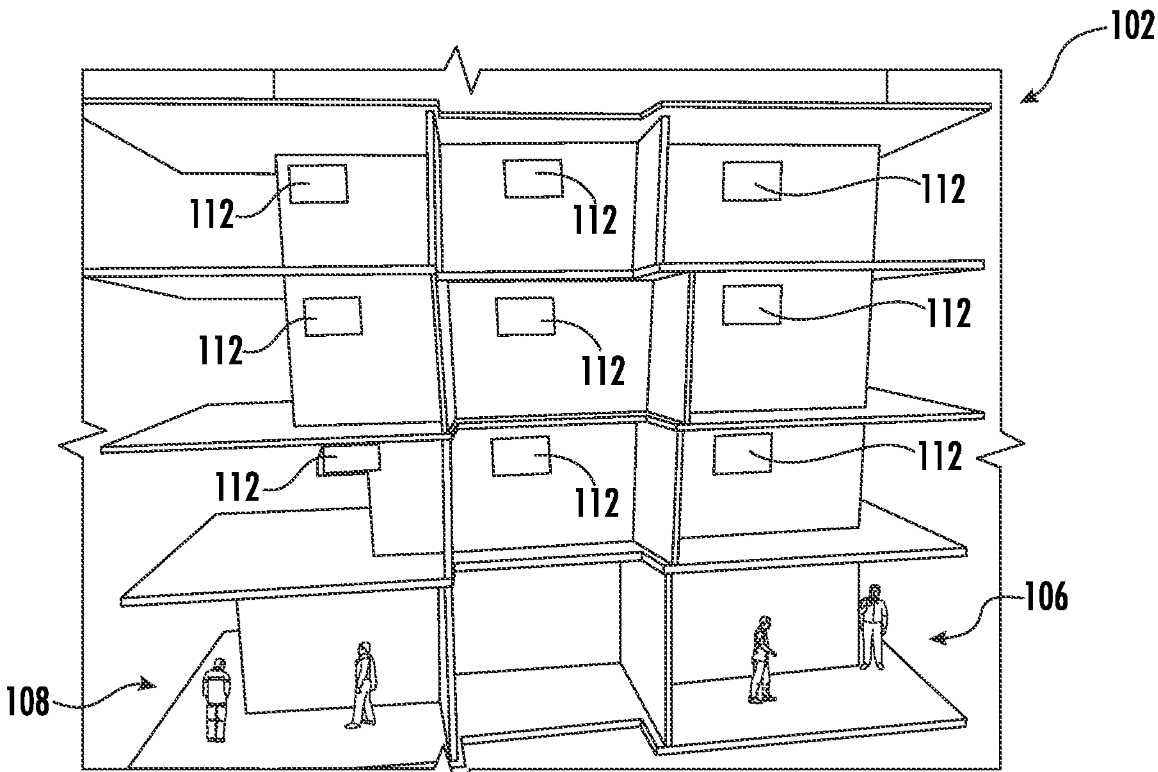


FIG. 1

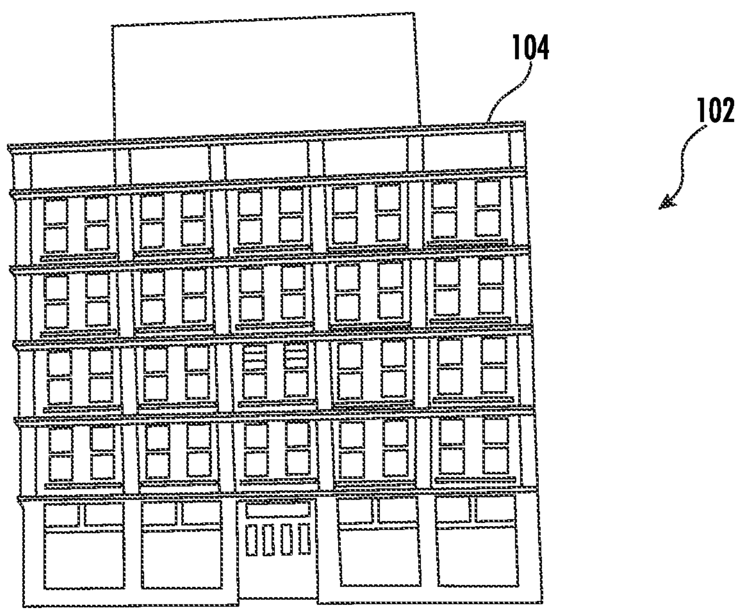


FIG. 2

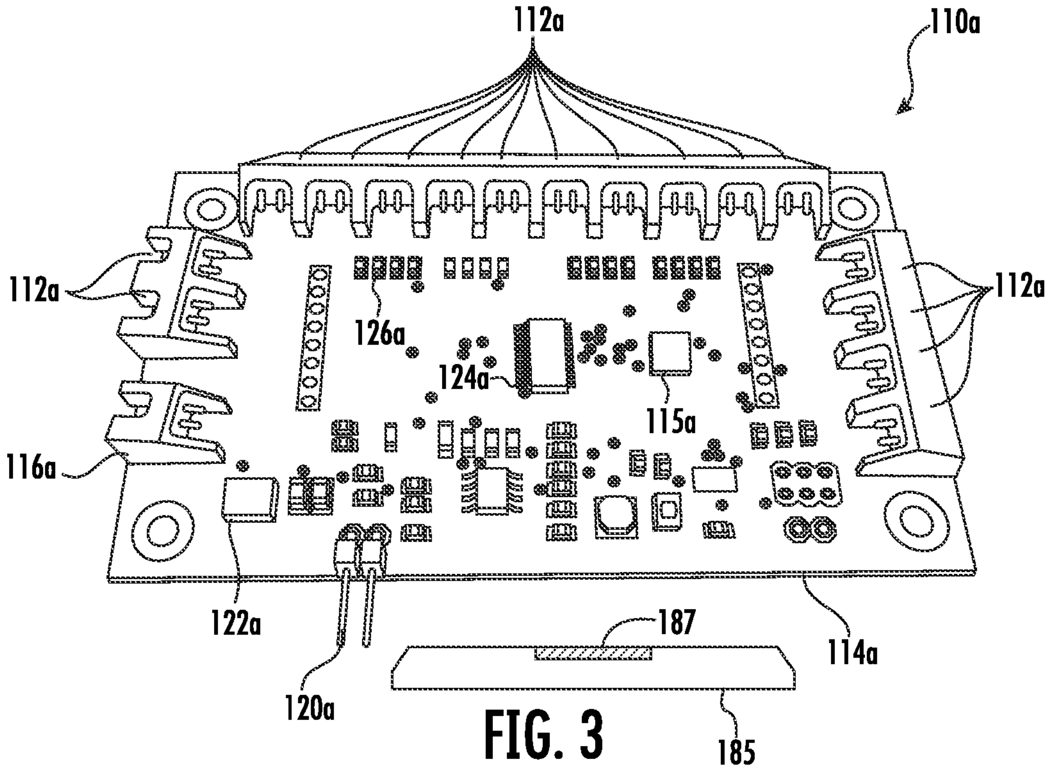

112a
110a
112a
112a
126a
124a
115a
116a
122a
187
114a
120a
185

FIG. 3

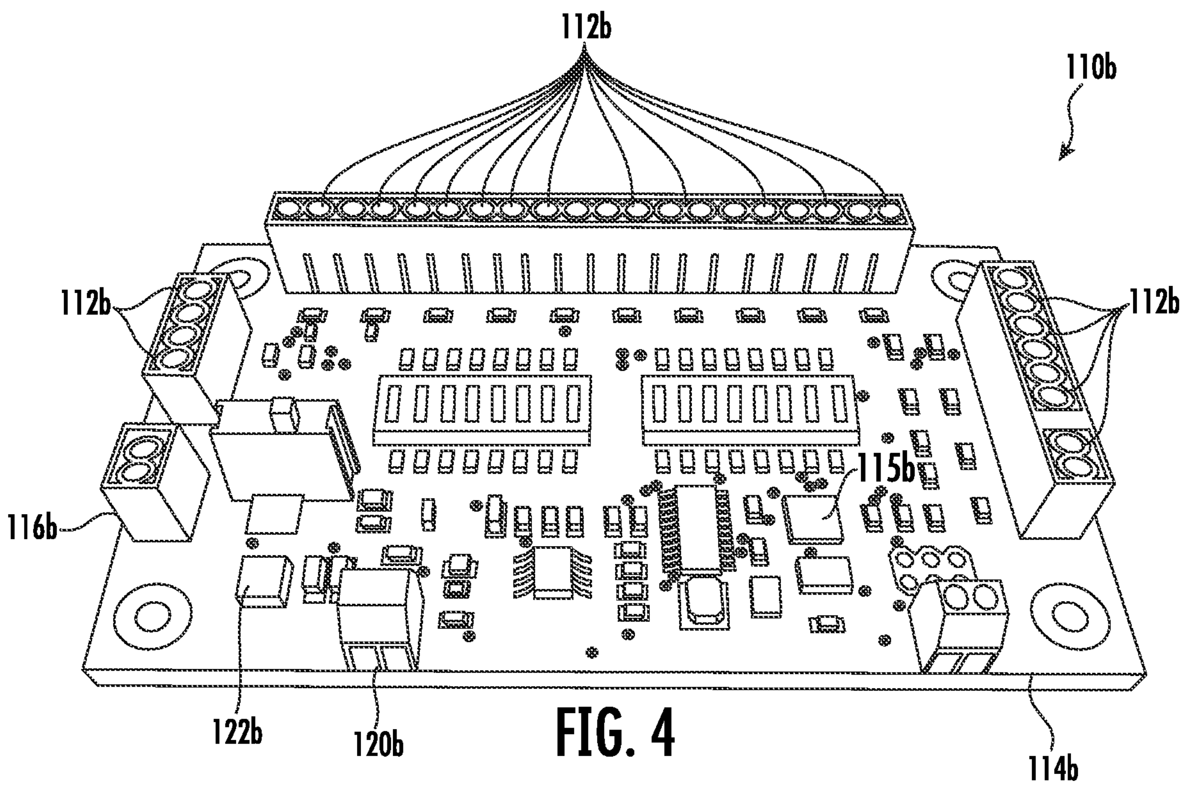

112b
110b
112b
112b
115b
116b
122b
120b
114b

FIG. 4

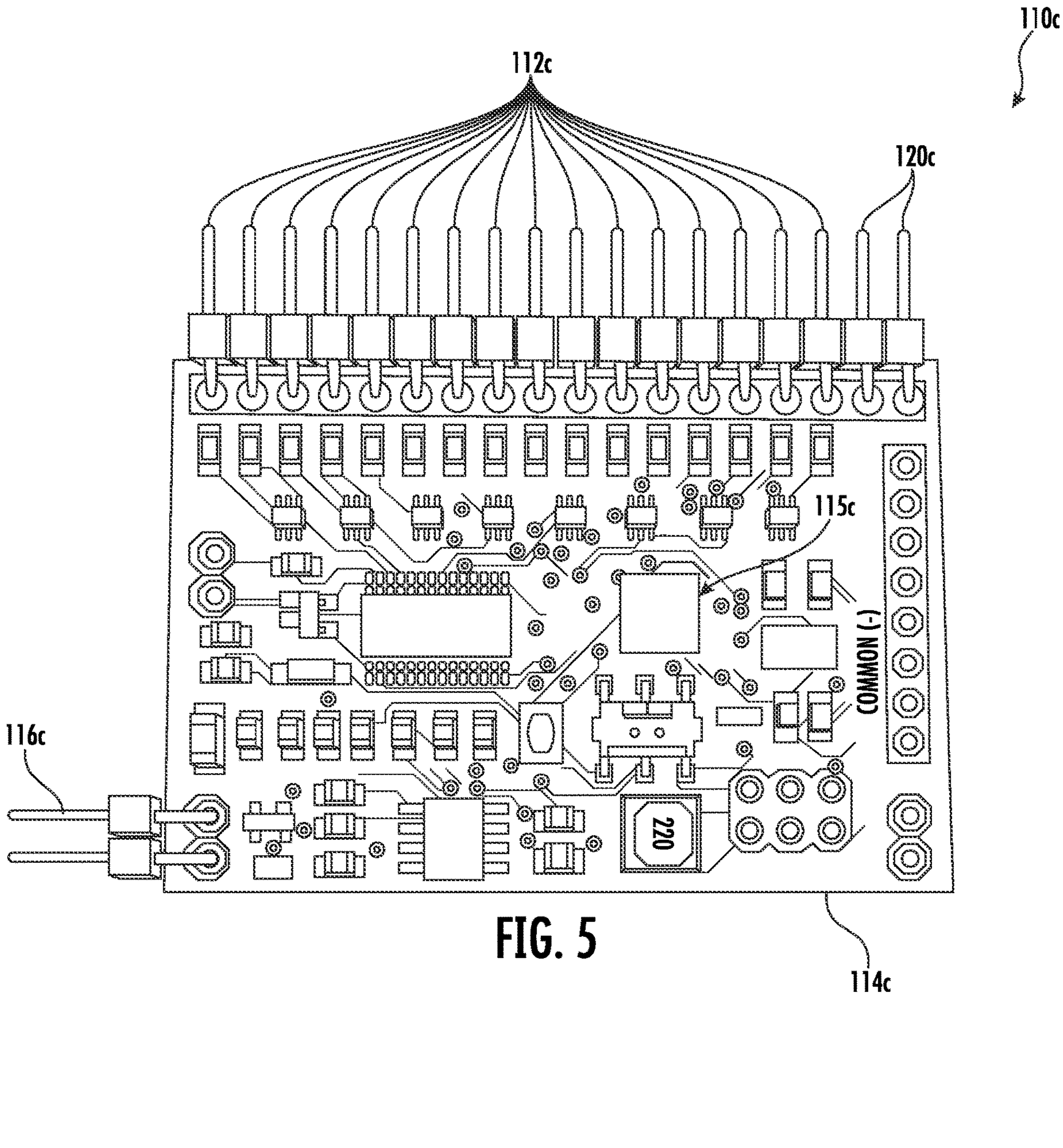


FIG. 5

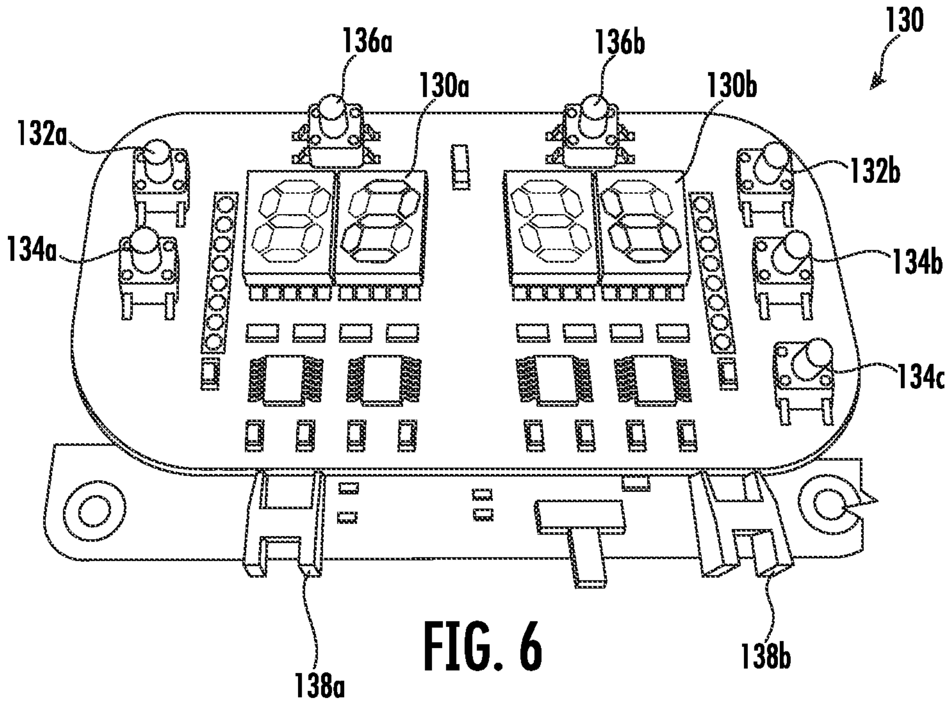

130
136a
136b
130a
130b
132a
132b
134a
134b
134c
138a
138b

FIG. 6

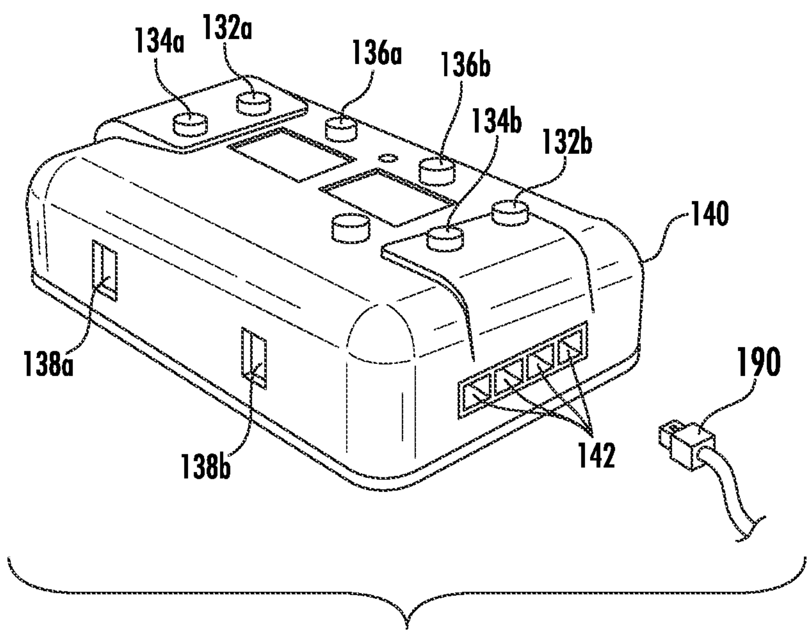

134a
132a
136a
136b
134b
132b
140
138a
138b
142
190

FIG. 7

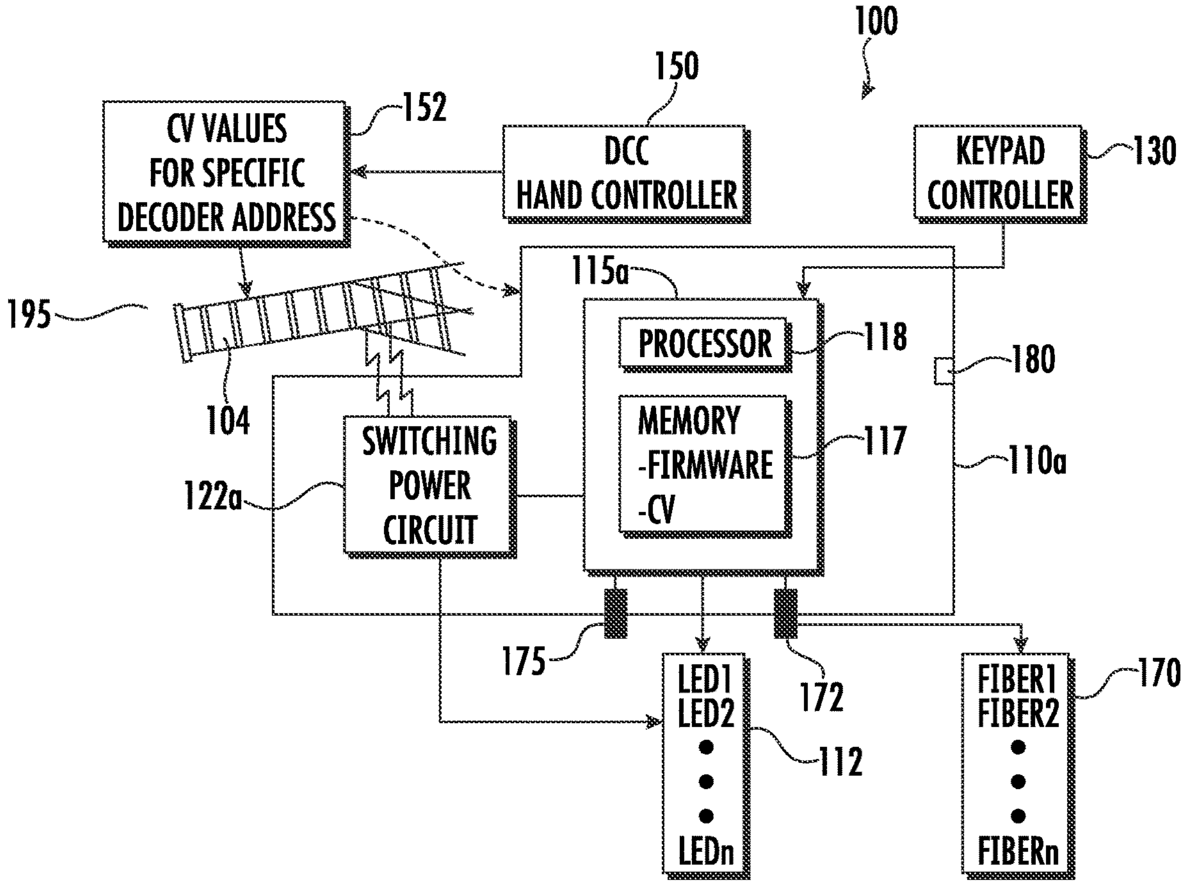

100
150
152
CV VALUES
FOR SPECIFIC
DECODER ADDRESS
DCC
HAND CONTROLLER
KEYPAD
CONTROLLER
130
195
115a
PROCESSOR
118
180
104
MEMORY
-FIRMWARE
-CV
117
110a
122a
SWITCHING
POWER
CIRCUIT
175
172
LED1
LED2
LEDn
112
FIBER1
FIBER2
FIBERn
170

FIG. 8

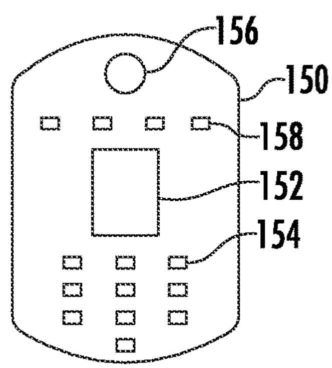

156
150
158
152
154

FIG. 9

200

202 204 206 208

	CV RANGE	VALUE	PORT NUMBER															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LED LIGHT LEVEL	128-143	0-99	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
ANIMATION MODE	144-159	0-18	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
TIME ON	160-177	0-255	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
TIME OFF	176-191	0-255	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
TIMER ON	192-207	0,1	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207

DISABLED	0
ALWAYS ON	1
FLICKER	2
BLINK	3
BEACON	4
FADE	5
RANDOM	6
STEP	7
FLASHING	8
TV SIMULATION	9
MARS	10
MIMIC F11	11
MIMIC F12	12
ARC WELDING 1	13
ARC WELDING 2	14
ARC WELDING 3	15
LIGHTNING 1	16
LIGHTNING 2	17
LIGHTNING 3	18

210

FIG. 10

```
const byte LEDLevel_CV          = 128;          // first value of CV range
const byte LEDMode_CV           = 144;          // first value of CV range
const byte time_on_CV           = 160;          // first value of CV range
const byte time_off_CV          = 176;          // first value of CV range
const byte LEDFadeMode_CV       = 192;          // first value of CV range
const byte trigger_CV           = 192;          // first value of CV range    For devices with INPUTS
const byte triggerDuration_CV   = 224;          // first value of CV range
const byte presetMode_CV        = 240;          // first value of CV range    // INTERNAL USE ONLY
// CV 144-159       0 = disabled
//                  1 = Always on               Activate with F5
//                  2 = flicker                 Activate with F5
//                  3 = blink                   Activate with F5
//                  4 = rotating beacon         Activate with F5; only setable via CV's
//                  5 = fade                    If set, command to turn it on/off will fade-Fade pin can't do anything else.
//                  6 = random                  Default MODE - Activate with F10 or F6 if not in Operation mode
//                  7 = Step                    Trigger with F0 - runs once on change
//                  8 = flashing                Activate with F10 - can create alternate blinking of two Odd/Even LEDs
//                  9 = TV Simulation           no configurable features
//                  10 = MARS
//                  11 = mimic pin 1            only function On/Off - since DT400 can't get to FN keys above 12
//                  12 = mimic = pin 2          only function On/Off - since DT400 cant get to FN keys above 12
//                  13-15 = Arc Welding
//                  16-18 = Lightning
//                  19 = TBD
//
////////// RAM ONLY VALUES ///////////////////////
bool            LEDstate[17];           // Ram only     indicates if the specific Port is on or off
byte            LEDfade[17];            // Ram only     stores the current PWM value to be incremented or decremented for Fade until zero.
unsigned long   trigMillis[17];         // Ram only     universal time trigger; every pin can be different
bool            function state[17]      // Ram only     used to store the state of the function Key on the controller to override animations
byte            beacon[17];             // Ram only     used to time & sequence the Rotating Beacon.
bool            beaconUP[17];           // Ram only     keep track of each beacon sequence separately
bool            LEDfadeUP[17];          // Ram only     keep track of whether the timer is in use or not.
bool            portState[17];          // Ram only     keep track of port state
bool            triggerState[17];       // Ram only     keep track of detector state
unsigned long   triggerTimeout[17];
bool            Accessory[17];          // Ram only     keep track of the Accessory State that mirror Triggers.
bool            functionkeyDetect[17];  // Ram only     simulate detection with function key in F5 animation mode
byte MAX_LED_LVL    = 50;               // for flash function only
byte MAXvoltage     = default_MAXvoltage;   // at this configuration 12 is 3.3V max for the 3V LEDs
```

FIG. 11

```
long          flashMillis              = 0;
bool          flash_sequence           = true;
byte          defaultFlashDelay        =50;
/////// These are the CV Addresses in EEPROM (122-127)   128-206 are array variables
const byte MFdecoder_CV                = 47;
const byte Product_CV                  = 48;
const byte ModeNumber_CV               = 49;
const byte fastBoot_CV                 = 50;
const byte MAX_Voltage_CV              = 60;        // CV for factoring downshifting the max PORT voltage. 45%-12v based on 25VAC input
const byte PowerSetting                = 61;        // the actual voltage of the input source 12v to 25v, default is 25
const byte keypad_installed_CV         = 62;        // CV for bypassing delay to clear keypad for NON keypad boards.
const byte defaultResetMode_CV         = 63;        // CV for setting what value the LEDMode[] is when reset - this is not reset by RESET
byte DCC_timeout_CV                    = 112;       // CV for storing second to wait until going into animation
byte flickerLOW_CV                     = 113;       // CV low end of the random range x10
byte flickerHIGH_CV                    = 114;       // CV high end of the random range x10
byte beaconspeed_CV                    = 115;       // CV for the speed of the dim up & down, simulating speed of rotation
byte fadeSpeed_CV                      = 116;       // CV location to save brightness in EEPROM
byte fadeTimeX_CV                      = 117;       // CV for seconds multiplier for Fade delay
byte random_delay_CV                   = 118;       // CV location to save brightness in EEPROM
byte random_percentON_CV               = 119;       // CV for determining what percentage of lights on vs Off for Random
byte randomModeOnePin_CV               = 120;       // CV sets the random to run only one pin at a time
byte randomModeFade_CV                 = 121;       // CV sets Random pins to Fade on and Off
byte step_delay_CV                     = 122;       // CV for LED RampUpDown speed
byte step_type_CV                      = 123;       // CV to switch STEP mode 0 Chase or 1 Race
byte flash_delay_CV                    = 124;       // CV for maximum brightness setting in EEPROM
byte LightningDelay_CV                 =125;        // CV for increasing the length between strikes. x 1 min. so 255 is 4.25 minutes
// CV 126-127 OPEN
bool blinking_LEDs             =false;
byte          ProgrammerCount = 0;                  // used to count 3 seconds on button 2 (horn)
bool          ProgrammerMode = false;               // sets the decoder to programming mode
long previousMillis = 0;
long BlinkInterval = 500;                           // used for programming indicator
long ResetTime = 8000;                              // - 5 seconds with the logic loops involved
long resetMillis = millis();
////////////////////
//   KEYPAD VARIABLES //
```

FIG. 11 CONT.

```
                }
        }
        else

        if (!ProgrammerMode && (func_F0 || func_F1))  //  if user presses once, save default brightness; main LED's flash to confirm
                {


////////// FUNCTION F3 SAVE     SAVE Mode


                no_other_functionkey = |func_F4 && |func_F5 && |func_F6 && |func_F7 && |func_F8 && |func_F9 && |func_F10;
                if (func_F3 && no_other_functionkey)          // SET PIN MODE
                {
                  int mode = (int) last_speed/speed_factor - speed_offset;
                  if (mode <= 18)            // only save and flash the pin if the mode selected by Speed is valid
                  {
                    LEDMode[pinSelected] = mode;
                    EEPROM.update(143 + pinSelected,mode);

                    if [(mode =5)]
                    {
                    time_ON[pinSelected]   =  var50_ON[1];
                    time_OFF[pinSelected]  =  var50_OFF[1];
                    fadeTimeX              =  var50_X[1];

                    EEPROM.update(time_on[pinSelected]-1+LED_port, time_ON[pinSelected]);       // CV 160-176
                    EEPROM.update(time_OFF[pinSelected]-1+LED_port, time_OFF[pinSelected]);     // CV 176-192
                    EEPROM.update(fadeTimeX_CV, fadeTimeX);
                    EEPROM.update(LEDFadeMode_CV,1);                                            // default single "S" is with timer
                    EEPROM.update(presetMode_CV-1+pinSelected,keyValue);                        // save 2-digit keypad mode setting
                    }


                    flashPin(pinSelected,5);
                }
        }

//////////FUNCTION F4 SAVE
        no_other_functionkey =     |func_F3 &&| func_F5 && |func_F6 && |func_F7 && |func_F8 && |func_F9 && |func_F10;
        if (func_F4 && no_other_functionkey)          // Save the brightness level
        {
                if (func_F0)        // ALL Pins SET
                {
                  for(uint8_t i=0;i<16;i++)
                    {
                    EEPROM.update(128+i, selectdimlevel);      // Save default LED level
                    LEDLevel[i+1] =   selectdimlevel;
                    }
                flashALL(4)                        // this ends with LED off

```

FIG. 12

```
//////////////////////////////
//     LEDMode = 1     //////////
/////////////////////////////


if (Animate_ON)
   {
   for (byte i=1;i<17;i++)
     {
               if (LEDMode [i] = 1)
               {

               if (portState[i] && LEDstate[i] = 0)          // if portState turned on and LED was off, turn it on
                         dimPin(i,LEDLevel[i]);


               if (!portState[i] && LEDstate[i] = 1)         // if LED is on and portState expired, turn it off
                         dimPin(i,0);
     }
   }
 }



/////////////////////////
//// F2 FLICKER ROUTINE ////
/////////////////////////




if (Animate_ON)
   {
     for(byte i=1;i<17;i++)
       {
               if (LEDMode[i] = 2)                    // LED on this pin is set to flicker
                 {
                         if (millis() > trigMillis[i])   // time has expired for this pin
                         {
                           if (LEDstate[i] - 1)          // LED is ON
                                 {
                                   dimPin(i, LEDLevel[i]);
                                   LEDstate[i] = o;      // forced for animation
                                 }
                           else                          // LED state if OFF
                           {
                                 dimPin(i, LEDLevel[i]/1.5);
                                 LEDstate[i] = 1;
                           }
```

FIG. 12 CONT.

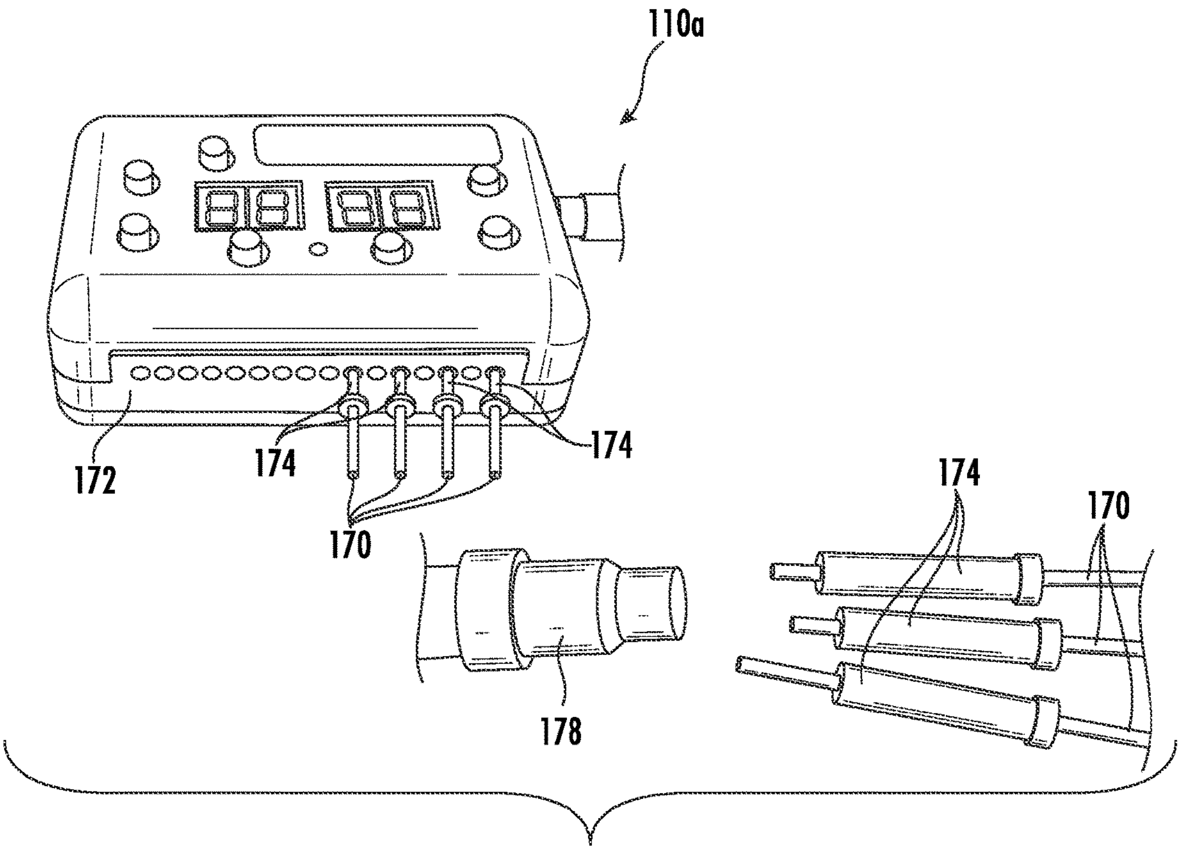


FIG. 13

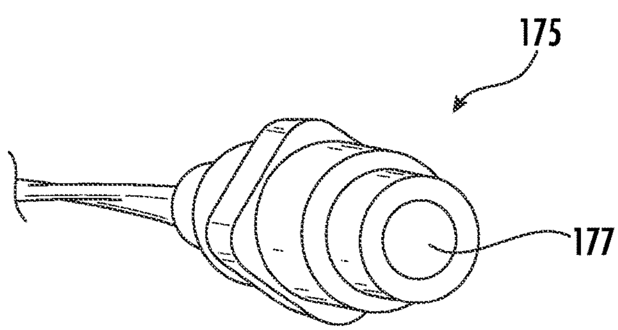


FIG. 14

LED SCENE CONTROLLER FOR A MODEL TRAIN SYSTEM AND RELATED METHODS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. provisional application No. 62/932,918 filed Nov. 8, 2019, which is hereby incorporated herein in its entirety by reference.

TECHNICAL FIELD

[0002] The present invention relates to the field of model trains, and, more particularly, to a LED scene controller for a model train system and related methods.

BACKGROUND

[0003] Model train systems, also known as model railroads, have benefited from the electronics age due to miniaturization and digital signal processing. One of the most dramatic changes has been in the past 20 years when it became possible to individually control multiple train engines on a common track. In the past a DC current was run through the two (or three) rails of the track so that more voltage resulted in more speed. To reverse the direction of the train, the polarity of the plus and minus voltage was reversed. A drawback was that this resulted in all engines on the same track would behave as one.

[0004] To provide for separate control of each train a method was developed that uses a pulsed DC current applied to the track. The time between pulses can be varied by milliseconds so there became a long pulse and short pulse. The timing of these pulses could be measured by a microprocessor to distinguish 1's and 0's. These were assembled into 8-bit bytes which then could be deciphered as commands and addresses. A board is installed inside a train with a microprocessor (MCU) having a decoder and software that interprets these bytes and in turn controls the voltage to a particular train engine motor making it go or not.

[0005] Included in the signal of bytes is a unique address (1 byte or 2 bytes) that must match a particular train in order for that train to follow the instructions that follow. In addition to motor function, each decoder may respond to multiple function commands (on or off). A decoder with these two controllable properties is generally identified as a multi-function decoder.

[0006] Furthermore, the same DCC system can be used to communicate with other remote devices that have a decoder but are stationary. Using the same bus as the track rails, signals can be sent to devices that control track switches (typically). The decoder uses the same addressing scheme so that a single address can be recognized and responded to. Since the track switch only needs to be thrown or closed (left or right) an accessory decoder decodes the address and then a single instruction (on/off, left, right) whereas the multi-function decoder used in the engine will respond to speed and up to 29 Function commands.

[0007] Notwithstanding advances in model train systems, there is a need in the art for an improved device to program and control effects of the miniature scenery that is installed around the railroad tracks such as lighting in buildings.

SUMMARY

[0008] An LED scene controller for a model train system is disclosed. The LED scene controller includes a printed circuit board (PCB) having a decoder and a plurality of light emitting diode (LED) ports mounted thereon, and a plurality of LEDs with each LED coupled to a respective LED port of the plurality of LED ports. The decoder includes a processor and a memory coupled to the processor, where the processor is coupled to the plurality of LED ports. In addition, the decoder is configured to receive a digital command control (DCC) signal comprising a plurality of configuration variables and to store the plurality of configuration variables in the memory.

[0009] The processor is also configured to read the plurality of configuration variables stored in the memory to control the plurality of LEDs coupled to the plurality of LED ports. The memory may include non-volatile memory that is configured to store software for operating the decoder and store the plurality of configuration variables. The decoder may also include a special effects component coupled to one of the plurality of LED ports and the processor configured to control the special effects component, and a switching power circuit configured to regulate power to the processor and the plurality of LEDs and special effects component.

[0010] The decoder is assigned a unique address by the DCC controller in order to program the decoder to receive and store only the plurality of configuration variables corresponding to the unique address. A first configuration value of the plurality of configuration variables may correspond to a brightness of the LED coupled to the respective LED port. In addition, the decoder is configured to automatically switch to an animation mode when no DCC signal is detected.

[0011] The DCC hand controller may comprise a keypad that is configured to generate the DCC signal comprising the plurality of configurable variables to control the plurality of LEDs. The switching power circuit may be configured to receive power from the model train track or an auxiliary power source.

[0012] In another particular aspect, an LED scene controller for a model train system is disclosed and includes a digital command control (DCC) hand controller, a model building, and a decoder having a processor coupled to a memory and a plurality of light emitting diode (LED) ports. The decoder is placed within the model building and is configured to receive a digital command control (DCC) signal comprising a plurality of configuration variables from the DCC hand controller and to store the plurality of configuration variables in the memory. The decoder includes a plurality of special effects components with each special effects component coupled to a respective LED port of the plurality of LED ports. The processor is configured to read the plurality of configuration variables stored in the memory to control the plurality of special effects components coupled to the respective LED port. In addition, the decoder includes a plurality of resistors coupled to the plurality of LED ports and is configured to adjust a voltage per LED port. The decoder also includes a switching power circuit that is configured to regulate power to the processor and the plurality of special effects components. The decoder is assigned a unique address by the DCC controller in order to program the decoder to receive and store only the plurality of configuration variables corresponding to the unique address. In addition, at least one of the special effects components comprises an LED, and a first configuration

value of the plurality of configuration variables corresponds to a brightness of the LED coupled to the respective LED port.

[0013] In another particular aspect, a method to control an LED scene controller for a model train system is disclosed. The LED scene controller includes a printed circuit board (PCB) having a decoder and a plurality of light emitting diode (LED) ports mounted thereon. The decoder includes a processor and a memory coupled to the processor, and the processor is coupled to the plurality of LED ports. The method includes receiving a digital command control (DCC) signal comprising a plurality of configuration variables from a DCC hand controller, and storing the plurality of configuration variables in the memory. The method also includes switching to an animation mode when no DCC signal is detected. In addition, the method includes assigning a unique address to the decoder by the DCC controller in order to program the decoder to receive and store only the plurality of configuration variables corresponding to the unique address.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. **1** is a perspective view of a building having an LED scene controller operating the lighting with the facade removed in which various aspects of the disclosure may be implemented;

[0015] FIG. **2** is a perspective view of the building of FIG. **1** with the facade installed;

[0016] FIG. **3** is a perspective view of the LED scene controller having a plurality of two pin outlets configured to individually control LEDs and other components;

[0017] FIG. **4** is a perspective view of another aspect of the LED scene controller having screw terminal adapters;

[0018] FIG. **5** is a perspective view of another aspect of the LED scene controller having terminal pins;

[0019] FIG. **6** is a perspective view of a keypad to program and configure the LED scene controller;

[0020] FIG. **7** is a perspective view of a case for the keypad and LED scene controller;

[0021] FIG. **8** is a block diagram of the LED scene controller;

[0022] FIG. **9** is a diagram of a DCC controller;

[0023] FIG. **10** is a chart of configuration variables that may be programmed to the LED scene controller;

[0024] FIG. **11** is an algorithm that may be executed by the LED scene controller to program the configuration variables;

[0025] FIG. **12** is an algorithm that may be executed by the LED scene controller for an animation routine;

[0026] FIG. **13** is a perspective view a fiber lighting manifold as part of the LED scene controller and fiber connector pins; and

[0027] FIG. **14** is a perspective view of a fiber lighting dongle.

DETAILED DESCRIPTION

[0028] The present invention will now be described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

[0029] There is currently a standardized system of digital command control ("DCC") for model train systems. Accordingly, it is common in the industry to read a signal from the railroad track and process it into usable code to drive a model railroad engine.

[0030] During the past ten years with the further miniaturization of component hardware it became more viable to include sound within the core functions of the decoders. In addition, software has also been developed to connect to the DCC system in order to monitor and control the actions of the trains on the layout from the computer.

[0031] One aspect of the invention disclosed herein for a new LED scene controller is directed to a new and improved system that allows configuration of various features without programming skills or a computer and specialty software. For example, the LED scene controller has the ability to configure more than 5 million possible combinations of special effects or animations (sound and motion). For one audience the LED scene controller will stand alone and be controlled and configured with non-DCC controls including via Bluetooth, Wi-Fi, etc. For another audience the LED scene controller can be controlled and configured by connecting to a DCC system. Both are possible and not mutually exclusive.

[0032] Referring now to FIGS. **1** and **2**, the LED scene controller has been installed in a model building **102**. The LED scene controller is coupled to a plurality of LEDs **112** that are placed strategically in the model building **102** as can be seen when the façade is removed as in FIG. **1**. On the bottom floor of the model building **102**, a scene **106**, **108** is staged on each side of the doorway to the model building.

[0033] The LED scene controller makes it easier to add LED lighting and other special effect components (i.e., animation) to miniature scenery or "scenes" as in FIG. **2** that are typical in model railroads than the prior controllers. However, the LED scene controller described herein is not limited in use to model railroading. Additional implementations can be any miniature scene, doll houses and even single function objects.

[0034] In a particular aspect, an implementation of the LED scene controller is inside N scale (1:160 scale) or HO (1:87 scale) buildings. The board of the LED scene controller is thin and small and allows the application to avoid running many wires from the scale building to a power source. All the wires (but two for power and control) remain contained inside the small building, for example.

[0035] Another challenge for hobbyists is determining what voltage/size LEDs to use and electrically understanding what value resistor is needed and the right power supply as LEDs do require a resistor between the power source. The LED scene controller solves this problem by including resistors on the LED scene controller board and allowing the adjustment of voltage per pin via the software stored in the LED scene controller. The LED scene controller is configured to operate from 12 to 25 Volt AC or DC, making it flexible.

[0036] Referring now to FIGS. **3**-**5**, the LED scene controllers described herein are exemplary and have similar components. Accordingly, the LED scene controller **110***a* in FIG. **3** has a plurality of two pin outlets configured to individually control special effects components such as

LEDs and other components. The LED scene controller **110***b* in FIG. **4** has screw terminal adapters, and the LED scene controller **110***c* in FIG. **5** has terminal pins. Although the LED scene controllers **110***a-c* have different configuration of the pins/ports, they function similarly. Thus, references to the LED scene controller **110***a* used in the description hereinafter will also apply to the LED scene controllers **110***b* and **110***c* as well.

[0037] The LED scene controller **110***a* includes a "DCC IN" pins **120** that are used to connect to the model train track or to a DCC controller. In addition, the LED scene controller **110** includes a PCB **114***a* where the components are mounted thereon such as a switching power supply **122***a*, a decoder **115***a*, capacitors **124** and resistors **126**.

[0038] With respect to special lighting effects, there are single board solutions for some of the effects. In those cases the effects are fixed. In contrast, the LED scene controller **110***a* allows for the selection and change of any of the effects on any pin **112***a*.

[0039] There are some effects that require multiple LEDs, for example two alternating blinking lights. Another is the "race" effect used to simulate airport runway approach lights. Another is a marquee effect. All of these are possible using the LED scene controller.

[0040] Despite the different "form factors" available for the LED scene controller **110***a*, they use the same basic electronic parts and design and perform substantially identical with respect to the effects and configuration and control via a DCC signal.

[0041] The LED scene controller **110***a* may be configured for the addition and synchronization of animation or unique sounds, for example, a light coming on in a saloon could trigger the noise of the saloon via an audio output. Another example that uses light, sound, and motion is a crossing gate and detection where the train on the track crosses a sensor that activates the gate to lower, lights to flash and bell to sound. The gate is controlled by a motorized actuator. Other sensors and/or scripting could cause a train announcement to be played once the train arrives at a station.

[0042] Multiple remote speakers could also be possible. Each pin **112***a* triggers a different background sound at a different location. The sounds may be downloaded and installed on a SD card via a web-based software and a PC. A library of sounds is available from 1000's of sources to use with the LED scene controller **110***a* or the user can record their own. USB adapters for SD cards already exist. The SD card with the stored sounds may be installed (slid into a slot) in the LED scene controller **110***a* and where the LED scene controller **110***a* is programmed to up to 16 separate sounds, one per channel to synchronized with the LEDs lights. For example, thunder and lightning can be programmed where multiple sound and lighting channels can be used to produce an exciting (multi-channel) visual and aural effect. Current solutions have one light and one soundtrack.

[0043] Continued miniaturization of electronics will likely drive the ability for more functionality. In addition, both Bluetooth and WIFI connections from a master console may be wirelessly connected to one or many LED scene controllers **110***a* scattered around a layout. This would allow the LED scene controller **110***a* to be untethered from DCC and from the base unit, which might not be connected to DCC, but would provide for computer control of the remote lighting.

[0044] The existing DCC systems have a number of slots where a slot is a memory space reserved in the DCC control system for each train engine. Some DCC systems have a few as 20 some as much as 500. Each LED scene controller **110***a* may use a slot if it is set as a multi-function decoder, otherwise as a switch there are a possibility of 2044 addresses and there is not a slot problem. Also, this is usually not a problem on small to medium layouts but on a larger setup this could be an issue.

[0045] However, the LED scene controllers **110***a* can be configured to the same address thus using only one slot. But with a bigger scenes and the desire to connect more controllable aspects, an alternative method for connecting and controlling the LED scene controller **110** is needed. A communication protocol such as one way broadcasting via Bluetooth or Wi-Fi to the LED scene controllers **110** connected allow this expansion.

[0046] Referring now to FIGS. **6** and **7**, a keypad to program the LED scene controller **110***a* is depicted. The keypad **130** is relatively small and easy to use. The keypad **130**, as shown in FIG. **6**, may be configured for the user to select, configure and control core LED scene controller functionality. Often times while operating the model train engines, it may be difficult for the engineer to dial in the correct LED port/effect. The keypad **130** has been installed within a housing **140** as shown in FIG. **7**.

[0047] The keypad **130** is intuitive to use for this purpose. The keypad **130** may include a first display **130***a* and a second display **130***b*. The user can use up buttons **132***a*, **132***b* and down buttons **134***a*, **134***b* in order to increase or decrease the value shown on the respective display **130***a*, **130***b*. Function keys **136***a*, **136***b* and **134***c* can be used to select the desired configuration values and addresses as necessary.

[0048] The keypad **130** can connect to the LED scene controller **130** using receptacles **138***a*, **138***b* and/or **142**. The keypad **130** is configured to program the LED scene controllers **110** to animate in sequence so that with the push a button, the saloon light will come on, for example, and/or other lighting and motion items nearby as well and the sound of the saloon starts. Multiple preconfigured "scenes" can be programmed so that a detector, magnet, IR (motion) or single button starts that animation.

[0049] In yet another aspect of the invention, a switch multiplexer may be included that would allow for the installation of switches or push buttons in remote locations whereby pushing the switch activates a pre-configured collection of LED scene controller ports/behaviors. Using the saloon example, a remote switch on a wall/fence protecting a scene or layout would allow a visitor to activate the scene without getting too close or disrupting any other activity on the scene or layout. In addition, a LCD-type display could be implemented to include more graphic indications and customizable screens for controlling the model train system.

[0050] Further, the function keys of a single decoder address may be used to address as many as 500 separate LED scene controllers via the master console. This provides for a simple pushbutton method to control the entire range of functionality via a DCC system or via a DCC support software application.

[0051] The LED scene controller **110** may have multiple special effects built into one unit. For example, this may include individual port LED brightness (voltage) setting, flicker, adjustable blink, rotating beacon, fade on and off, random, chase, race, step, alternating flashing, television simulation, mimic, arc welding, lightning, and switch relay. In addition, the level of brightness for each pin (also referred to interchangeably herein as "port") LED can be set. The effect can be adjusted and selected for each pin individually via the keypad **130**. Moreover, all the functionality may be configured and controlled via an existing DCC system hand controller **150** as shown in FIG. **9**, the keypad **130**, or a computer via a USB.

[0052] Referring now to FIG. **8**, each of the LED scene controllers includes an on board DCC decoder. This allows each LED scene controller to have its own address, which means that each LED (or other animation component) connected to each LED scene controller can be controlled separately and each LED scene controller can be controlled separately.

[0053] The LED scene controller may be powered by DCC track power or via a multitude of AC and DC power supplies from 12-25V AC/DC. A mechanical switch allows the user to choose between configurations: 1) DCC powered only, 2) AUX power for the LEDs and DCC signal only in (avoids using DCC power), or 3) AUX power only; board switches to animation mode after a configurable time delay.

[0054] Different voltage LEDs can be used on any port with each port having a different setting. The LED scene controller may include two different sockets for power: 1) JST sockets (Japan Standard Terminal), and 2) 1.3 mm barrel plug.

[0055] The LED scene controller does not require soldering, wiring or programming skills to configure. Therefore, the case surrounding the electronics, for non-technical people, provides a superior experience and makes them feel the product is easy to use. The case for the LED scene controller may include a mounting bracket for positioning anywhere on the layout and easy removal.

[0056] The LED scene controller includes an LED driver and a MCU with at least 16 mHz clock to obtain real-time performance. The button design/use provides easy learning in less than one minute. In addition, the case and keypad are also removable and the internal main board can be purchased, installed and operated separately from the case and keypad. The LED scene controller is configured to operate fully without the keypad.

[0057] With the keypad installed, each LED port can be selected and configured with any one of 100 pre-set behaviors (special effects). For example, each LED port brightness can be set and saved using the keypad. The LED scene controller can switch between animation mode (active) and configuration mode with the push of a single button on the keypad. As explained above, the LED scene controller can be programmed with the keypad (or a computer via USB), and then the keypad can be disconnected.

[0058] The LED scene controller comprises a DCC detect logic so that it automatically goes into animation mode if no DCC signal is detected. When the DCC signal is turned on or connected, the LED scene controller is configured to act upon any further DCC instructions via the function keys.

[0059] The LED scene controller may be relatively small, for example, 50 mm×30 mm×1.0 mm, so that it can be placed inside very small enclosures. By placing the LED scene controller inside a building in order to individually control a "room by room" LED or other component, only two wires are needed from the building to a DCC power source. This greatly reduces the wiring burden, which can be significant.

[0060] The LED scene controller may include a two wire pause capability via a connection terminal on the main board. This allows a simple SPST switch (or momentary) to be connected to pause and turn off all the LED scene controller functions. When deactivated, the LED scene controller resumes animation mode.

[0061] The LED scene controller board may be configured for various connection configurations (JST plugs, screw terminals solder pads and solder holes, pins) to accommodate a multitude of installation requirements. Adapter test and connection boards can also be used to test and configure the board before it is installed in its final destination.

[0062] In a particular aspect, the system uses LED "chips". The LED chips are made specifically for the system. These are surface mounted LEDs that would melt if soldered. The LED chips are attached to a 9 mm×9 mm or smaller PCB with two solder pads. The LED chips are relatively inexpensive but cannot be hand soldered. The flat PCB surface provides an easy way to attach them to a structure/surface to be lit. The LED chips are reliable and inexpensive so that more LEDs can be used per dollar.

[0063] In addition, the LED scene controller comprises an onboard switching power circuit. The LED scene controller may be configured to activate either 5V DC or 12V DC mechanical relays off the LED ports. This allows for the control of objects, lights, and motors that need to be electrically isolated or require more amperage than the LED scene controller board supports.

[0064] The LED scene controller may include a 6-hole pin connection or USB connector that allows for in-shop software updates. Accordingly, the LED scene controller can be easily updated with the latest version and functionality. Virtually any LED from 3 v to 12 v can be used with the system and the DCC decoders of the LED scene controller comply with the NMRA DCC Standards.

[0065] Referring now to FIG. **8**, a block diagram of the system **100** is illustrated. The system **100** includes the LED scene controller **110*a*** having a decoder **115*a*** that comprises a memory **117** and a processor **118**. The memory **117** stores firmware that contains instructions for the processor **118** to execute. The memory **117** also stores the configuration variables (CV) that correlate to how the LEDs **112** (and other special effect components) are programmed to function. The keypad controller **130** may be coupled to the LED scene controller **110** to program the CVs as explained above.

[0066] A DCC hand controller **150** can also be used to program the CVs for the LED scene controller **110*a*** via a DCC signal that is transmitted over the model train track **104** or directly to the LED scene controller **110**. A capacitor **124** may be included with the LED scene controller **110*a*** in order to rectify any power interruptions and to eliminate any flicker in the operation of the LEDs **112**.

[0067] Referring now to FIG. **9**, the LED scene controller **110** may be configured via the DCC hand controller **150** as explained above using configuration variables. The hand controller **150** includes a display **152**, a keypad **154** for entering values, a speed control **156** and function keys **158**.

[0068] A chart of configuration variables (CV) that may be programmed to the LED light board is shown in FIG. **10** and generally designated **200**. The first column **202** describes the behavior of the LED, such as "LED light level." Moving to the next column to the right **204**, a "CV Range" is provided to correlate with a port number **208** of the LED scene controller **110**. A value that may be assigned for the behavior of the LED is provided in column **206**, for example 0-99. The unique address of the desired LED scene controller to be programmed must be set in order for the decoder of the desired LED scene controller **110** to read the DCC signal and store the CV values being transmitted. In addition, in table **210**, values that correlate to a particular type of animation are provided. For example, a value of 3 for the "Animation Mode" would program the LED coupled to a particular port to "blink."

[0069] Another feature is a collection of 100 preset Animations. Rather than having to enter discrete values for blink time on or off, the QuickSet table allows complex configuration for new users.

[0070] Algorithms that perform the functions are disclosed in FIGS. **11** and **12**. For example, FIG. **11** provides algorithm for programming configuration values and FIG. **12** discloses an animation routing.

[0071] Referring now to FIGS. **13** and **14**, the LED scene controller **110** can also be used with fiber optical fiber cable lighting **170** (FIBER1, FIBER2 . . . FIBERn). Instead of plugging in an LED via a plug and wire, individual fiber cables **170** can be inserted into a fiber lighting manifold **172** as part of the LED scene controller **110***a*. The LEDs are permanently located inside the controller **110***a* and the manifold **172** is aligned with the end of the fiber cable **170** with the LED. This allows individual fiber cables to have the same animation behaviors as described above.

[0072] Specially designed fiber connector pins **174** allow the insertion of different cable sizes from 0.25 mm to 2 mm. The manifold **172** is tapered and having receptacles **178** to cause the pin **174** to compress the inserted fiber cable and to hold the cable in place in the manifold **172**. It is a friction fit and the pin **174** can easily be removed but will not fall out on its own.

[0073] A fiber lighting dongle **175** is also disclosed. An LED **177** is inserted into a plastic casing ~3 mm round. The LED **177** is plugged into a port/pin connected to the decoder **115***a*. Using one of the fiber lighting pins, a fiber cable is connected to the dongle **175**. This allows lighting and controlling a single fiber light. Optical fiber cable is useful for certain lighting conditions and difficult to reach or place even with the smallest LED.

[0074] The controller may also be configured as an NMRA Accessory model. The decoder **115***a* can be switched by the user to operate as a multifunction decoder or an accessory decoder. The LED scene controller **110***a* can react to up to sixteen unique switch addresses when in the accessory mode. Any port can be assigned to a single switch (accessory) address or multiple addresses and ports. In addition, multiple controllers can respond to the same address or overlapping banks of address.

[0075] In another particular aspect, the LED scene controller **110** includes a USB port **180** for programming the animations via computer. The addition of computer connected programming allows for expanding of functionality to scripting. Scripting allows a sequence of instructions to activate any of the animations on any port for any length of time. Multiple scripts can be run simultaneously on a controller. Each unit may have a unit serial number that is automatically generated. The LED scene controller **110***a* is configured to be connected to the computer via USB cable. Software on the computer is configured to control a plurality of aspects of the LED scene controller **110***a*.

[0076] A specialty "switch" adapter **190** can be plugged into an outlet of the LED scene controller **110***a* such that an "ON" signal activates a solid-state switch that in turn could activate other devices. There is also MODE setting on the controller **110***a* called "relay". When a port is set to this setting it will send 5V+ to the switch as prescribed by the timing set on the LED scene controller **110***a*.

[0077] The LED scene controller **110***a* may also be configured to allow the playback of sound files synchronized with animation. Sound files can be stored in memory on the controller **110***a*. The LED scene controller **110***a* may be configured so that the user can change or add sounds from their computer to the controller **110***a*. The LED scene controller **110** may be configured to be controlled independently from the DCC, detection circuits or via the script of a controller. The LED scene controller **110***a* can be configured only for sound, or the controller **110***a* can be configured for both controlling sounds and lighting and all can be configured using DCC.

[0078] Detection circuits **195** are relatively small and can slide under the model track **104** and can be either magnetically activated or IR. In operation, a passing train triggers the detector circuit **195** that activates a gate crossing, turns on a fiber optic crossing sign, turns the LED light at the station on, plays multiple sounds like the ding-ding of the crossing bell, train horn and stations sounds. Accordingly, various combinations of lighting, detection, motion, animation, control and sound is combined into a single controller.

[0079] A specialty power mounting base **185** may include magnets **187** to convey low voltage power to the controller. The power supply may be connected to the base instead of plugged into the LED scene controller **110***a*. In addition, a specialty POWER and LED connection mounting unit is configured so that the controller can be removed without unplugging. Magnets may connect power to each LED output. Thus, the LED scene controller **110***a* can be "popped" off, brought to the desk for programming and then snapped back at the layout location.

[0080] Many modifications and other embodiments of the invention will come to the mind of one skilled in the art having the benefit of the teachings presented in the foregoing descriptions and the associated drawings. Therefore, it is understood that the invention is not to be limited to the specific embodiments disclosed, and that modifications and embodiments are intended to be included within the scope of the invention.

That which is claimed is:

1. An LED scene controller for a model train system comprising:

a printed circuit board (PCB) having a decoder and a plurality of light emitting diode (LED) ports mounted thereon; and

a plurality of LEDs with each LED coupled to a respective LED port of the plurality of LED ports;

the decoder comprising,

a processor and a memory coupled to the processor, the processor coupled to the plurality of LED ports;

wherein the decoder is configured to receive a digital command control (DCC) signal comprising a plurality of configuration variables and to store the plurality of configuration variables in the memory.

2. The LED scene controller of claim **1**, wherein the processor is configured to read the plurality of configuration variables stored in the memory to control the plurality of LEDs coupled to the plurality of LED ports.

3. The LED scene controller of claim **2**, wherein the memory comprises non-volatile memory configured to store software for operating the decoder and to store the plurality of configuration variables.

4. The LED scene controller of claim **3**, further comprising a special effects component coupled to one of the plurality of LED ports and the processor configured to control the special effects component.

5. The LED scene controller of claim **3**, the decoder further comprising a switching power circuit configured to regulate power to the processor and the plurality of LEDs.

6. The LED scene controller of claim **3**, wherein the decoder is assigned a unique address by the DCC controller in order to program the decoder to receive and store only the plurality of configuration variables corresponding to the unique address.

7. The LED scene controller of claim **3**, wherein a first configuration value of the plurality of configuration variables corresponds to a brightness of the LED coupled to the respective LED port.

8. The LED scene controller of claim **6**, wherein the decoder is configured to automatically switch to an animation mode when no DCC signal is detected.

9. The LED scene controller of claim **8**, further comprising a keypad configured to generate a DCC signal comprising the plurality of configurable variables to control the plurality of LEDs.

10. The LED scene controller of claim **5**, wherein the switching power circuit is configured to receive power from the model train track or an auxiliary power source.

11. An LED scene controller for a model train system comprising:

a digital command control (DCC) hand controller;

a model building; and

a decoder having a processor coupled to a memory and a plurality of light emitting diode (LED) ports, the decoder placed within the model building and configured to receive a digital command control (DCC) signal comprising a plurality of configuration variables from the DCC hand controller and to store the plurality of configuration variables in the memory.

12. The LED scene controller of claim **11**, further comprising a plurality of special effects components with each special effect component coupled to a respective LED port of the plurality of LED ports.

13. The LED scene controller of claim **12**, wherein the processor is configured to read the plurality of configuration variables stored in the memory to control the plurality of special effects components coupled to the plurality of LED ports.

14. The LED scene controller of claim **13**, further comprising a plurality of resistors coupled to the plurality of LED ports and configured to adjust a voltage per LED port.

15. The LED scene controller of claim **14**, the decoder further comprising a switching power circuit configured to regulate power to the processor and the plurality of special effects components.

16. The LED scene controller of claim **15**, wherein the decoder is assigned a unique address by the DCC controller in order to program the decoder to receive and store only the plurality of configuration variables corresponding to the unique address.

17. The LED scene controller of claim **13**, wherein at least one of the special effects components comprises an LED, and a first configuration value of the plurality of configuration variables corresponds to a brightness of the LED coupled to the respective LED port.

18. A method to control an LED scene controller for a model train system, the LED scene controller comprising a printed circuit board (PCB) and having a decoder and a plurality of light emitting diode (LED) ports mounted thereon coupled to a plurality of LEDs, the decoder comprising a processor and a memory coupled to the processor, the processor is coupled to the plurality of LED ports, the method comprising:

receiving a digital command control (DCC) signal comprising a plurality of configuration variables; and

storing the plurality of configuration variables in the memory.

19. The method of claim **18**, further comprising switching to an animation mode when no DCC signal is detected.

20. The method of claim **18**, further comprising assigning a unique address to the decoder by the DCC controller in order to program the decoder to receive and store only the plurality of configuration variables corresponding to the unique address.

* * * * *