

LED Image Painter

Introduction

History

This design was inspired but not copied from the PixelStick commercial product. I have not owned, used or even touched an official PixelStick, I haven't even read their manual. I did briefly look at the marketing material on their website to get an idea of how it worked. I was also inspired by the light wand developed by photographer Michael Ross. I started out building one of these. It used the Mega2560 Arduino and a 2-line display with some buttons. I started adding features and quickly ran out of memory. I also wanted to support a second 1-meter LED strip. At this point I decided to do a total rewrite and use the newer much faster esp32 modules. This let the feature list grow and additionally the strip display speed was increased dramatically.

Using with Your Camera

Put your camera on a tripod and set it for a long enough time exposure so the shutter is open for the time it takes to display the image. Stand at your chosen starting position and open the camera shutter. Press the Image Painter start button and walk from right to left (looking at the camera), or left to right if you prefer, it is adjustable from the menus. Walk smoothly or erratically for the effect you want and hold the wand in the same position or wave it around. Use your creativity and see what kinds of images you can create.

Description

The Image Painter consists of the following major parts. The parts are all connected together into a single unit. The optional second LED strip is easily removed so the system fits into a smaller space.

1. LED strip. This is an aluminum extrusion with 144 LEDs and a diffuser cover. The LEDs are available with an adhesive backing that is used to stick the strip inside the extrusion.
2. Fixed Handle. The rotary control button is on the handle which is used to hold the Image Painter. The button is under your thumb and can be pushed and rotated.
3. Rotating Handle. This is useful when two LED strips are attached. It allows the user to spin the light stick.
4. Control Unit. This box holds the ESP32 computer that runs the LED strips and reads the rotary button switch.
 - a. OLED Display. The ESP32 module used includes a small OLED display that is used to show the file selections. It is also used to show menus so that the various settings can be changed.

5. USB Battery. This is the rechargeable battery that runs the control unit and the LED strips.
6. Roller Encoder Wheel. This wheel is used to control the display speed while rolling on the ground. It will allow for very exact image timing. The good news is the software supports it, the bad news is that the hardware mount for the wheel has not been designed yet.

144 or 288 pixels

The Image Painter uses one or two WS3812B pixel strips that have 144 pixels per meter. In actuality each meter strip is made from two 72 pixel strips. This results in a slight gap in the middle that can be seen in some images. When the second strip is installed the power wires make it difficult to get the pixels really close. This also results in a dark gap in the middle of the two strips that can be seen with some images.

Limitations

At full brightness each strip uses 43 Watts when all the LEDs are lit. Since the USB battery can only supply 35 Watts it is obvious that the display cannot be used at full brightness. However, some images do not have all of the LEDs lit at the same time so the brightness can sometimes be set to full power. For example, if the strip never has more than half of the LEDs on then the power consumption is only 21 Watts. This will work. In practice however I have found that due to the long night exposures it is seldom necessary to go brighter than about $\frac{1}{4}$ power.

Processing BMP Images

The images can be either 144 or 288 pixels in height. They must be rotated 90 degrees CCW and saved as 24bit color in Windows byte order. They can be as long as desired in width.

Examples

Here are a few example photos created with the Image Painter. People more creative than me will create much more interesting images. I'm including these just to illustrate a few simple examples. You can also go search the web for PixelStick images to get more inspiration.

Button Operations

All operations can be controlled by the rotary push button on the handle, this makes the image painter usable with one hand. The display shows the options and settings and filenames. The dial can be rotated left or right to choose items. The button is clicked to select the current item. A long press is used to access other options and is done by holding the button down for about a $\frac{1}{2}$ second or longer. This long press time is also adjustable if a longer or shorter time is preferred. The system has two modes of operation. After booting the display shows the currently

selected file. Rotating the dial left and right will move through the files on the SD card, or the internal patterns if they have been selected. Clicking the button starts the LED strip display. At this point the user walks around with the wand or waves it around depending on what effect is desired. The display shows the progress of the file and how much time is remaining. Settings are available to control the display speed, repeating the file, or even automatically skipping to the next file. The image can even be played twice, first forward and then backward for a mirror like effect. There is a chain option that will play all of the files from the current selection to the final file in the current folder. The brightness of the LEDs are of course also adjustable to balance the lighting with the rest of the scene.

To adjust settings a long press is used which will open the last menu that was used. A long press while in the menus returns to the normal play mode. Each menu except the main menu has an entry to return to the previous menu.

File Control Settings

There are more details below in the section on menus but here is a quick list of the things that can be changed for controlling the image display.

1. LED brightness.
2. Color Balance. Set the white balance for special effects or to correct color casts.
3. Gamma Correction. Adjust the display to match closer to our vision.
4. Frame hold time, how long each frame is displayed. Note that even when set to 0 there is a minimum frame time which is determined by the amount of time used to load the LED's. See the menu section below for more details. Frame advance can also be set to manual so the frame only changes when the button is pressed or the dial is rotated.
5. Fixed display time. It is also possible to have the system calculate the frame hold time. You just set how long in seconds you want the image to take and it will set the correct frame time based on the image size.
6. Reverse. So the user can walk from left to right or vice versa.
7. Upside Down. This allows the user to hold the wand high in the air without modifying the image file.
8. Pixel Double. This uses two pixels for each one so a 144 pixel image can be displayed on the entire 288 pixels when two strips are installed.
9. Scaling. This will reduce a 288 pixel image to 144 pixels.
10. Mirror. Plays the file twice, first forward and then in reverse.
11. Chain. Plays all the files from the current file to the last file in the current folder. Files are always in alphabetical order.
12. Start Delay. A wait after pressing the button until the file starts. All delays are specified in 1/10 seconds.
13. Repeat Count. How many times to repeat a file.
14. Repeat Delay. A wait in between file repeats.
15. Chain Repeat. How many times to repeat a chain.
16. Chain Delay. A pause in between chain repeats.
17. Built-In or File. Choose files from the SD or from a set of built in patterns.

18. Change settings for various built-in patterns.
19. Select Folder. Choose a folder on the SD card to look for BMP files.
20. Display Settings. Control some menu operations and display options.
21. IPC. Manages files that change settings for folders and files. These setting files are loaded when a folder is selected or when a file is run. The settings for a file revert to the original values after running the file. These files are also text and can be edited if desired for special operations. The file version will apply the settings when the named file is displayed. The folder version is loaded when a folder is selected.
22. Saved Settings. Change the settings that can be optionally loaded at boot time. It is also possible to save the current settings and manually load them when needed.
23. Macros. There are ten macro slots (0 to 9) available. Each of these can either load all of the settings or run a recorded sequence of operations. The record command records the images as they are played and the same sequence can then be played again using the run command. The save command stores the current settings which can then be restored with the load command.
24. BLE. Enable BlueTooth. There is an Android app that can monitor and control the Image Painter. An iOS one may come later.
25. Reboot. This will reset the system. It can of course also be done by unplugging the USB power supply.

Rotating and Clicking Button

The handle button is rotated and clicked to select different files, menu items, and to change settings. In the menus boolean values toggle with each click. Integer values go to another screen that shows the minimum and maximum values. Rotating the dial right increases the value while rotating to the left decreases it by the increment amount. The increment amount can be changed by clicking the button which will cycle between different values. One of the values will be a reset to the original value, activated with a long press. A long press is also used to accept the new value and return to the menu. A * indicates the active menu line. A + in front of the menu means another menu will be activated. The "Return..." menu goes back to the previous menu when clicked. A long press will return to file select/run mode. The current menu entry is remembered so that it will return to the same place when the long press is done from the select/run screen.

In select/run mode the current file is displayed on the top line. Optionally the next three files will be displayed below. Rotating the dial right moves the next file to the top line and rotating left moves the previous file back. By default it will stop at the first and the last file. An option allows rollover, so going left from the first file moves to the last file and moving right at the last file will display the first file again. This option also causes menus to behave similarly.

When a file is running after clicking the button the top of the display will show the progress of the file. The current filename is displayed. The line under the file will show the remaining seconds for the file. The rest of the screen will show the various delays, repeats, and chain information when those options are activated.

Button Menu Operations

Left/ Right

- Select menu
- Select file
- Inc/ Dec integers

Long press

- Open menu system, goes to the remembered menu point
- Return to run mode, the current menu point is remembered
- Accept integer

Click

- Run file when in run mode
- Select menu item, including return to previous menu
- Toggle booleans

Menu Details

Following is a detailed list of all the menu entries and what they do. Menu items that open another menu (a submenu) start with a '+'. Most submenus also have a first and last menu entry that returns to the menu above. A '*' is shown in front of the active menu line. The top menu will have a '-' indicating that there are menu items that are not visible due to scrolling. Just keep rotating the knob to the left and they will appear. Similarly the last entry on the screen will have a '_' if there are more menus below. Scroll right to make them visible as the '*' indicator moves down.

The submenu details are also described below after each menu item.

Main Menu

Switch to Built-Ins/Switch to SD

Clicking this toggles between running files from the SD card or using the built-in patterns.

File Image Settings

This submenu has the settings that control display attributes of the image files. The frame hold time is also used by some of the built-in image functions.

Timing Type (Column or Image)

This is a Boolean that switches between using a set frame hold time for each column of the image file or a calculated time based on the desired total display time. It is sometimes more convenient to set the total time that the image file is displayed. This mode will automatically

calculate the frame hold time such that the image will display for the chosen time. It is possible to request a time that is too fast for the image painter. If the calculated frame hold is less than about 4 mS the software cannot read the SD card fast enough to keep up. The image painter will try to do the best it can and the actual time will be displayed on the screen as the image plays.

Frame Hold

When frame time is selected this is the time in mSecs (see note below) that each line of information from the BMP file is displayed on the LED strip. It is commonly used to adjust the walking time while the file is being displayed. It is only applicable when Frame Advance is set to auto.

Note: This time is actually added to the minimum time that it takes to update the LED's in the strip. Each bit of information takes 1.2 uS and each LED needs 24 bits. If there are 144 LED's then the total time to load the data is $1.2 \text{ uS} \times 24 \times 144 = 4.32 \text{ mS}$. There is also a reset time at the end that loads the data into the LED's of >50 uS. Some brands require up to 1 mS so it is safest to assume that the column update time is about 5 mS. This time is added to the time set in the frame hold value, I.E. a hold value of 0 will still have a 5 mS frame time. There might also be additional time required for reading slower SD cards. Also be aware that displaying the file in reverse will slow down due to reading the SD file in the reverse order. A 900 column image would in theory take about 3.9 S. In testing we have found that reading the SD card actually slows this down to about 6 Seconds. Work is being done to try and improve this.

Fixed Time

When fixed time is selected as described above in **Fixed/Frame Time** this menu item allows the time to be set in seconds that the image will be displayed. The necessary frame hold time will be calculated and used while displaying the image file. This value will not be saved in the frame time so if frame hold is selected it will show the original value.

Frame Advance

The options are "auto" and "manual". In auto mode the frame hold time is used. In manual each line is displayed until the button is pressed, the dial is rotated, or the wheel pulse counter closes the contact. If the frame pulse counter is zero then the dial is used to advance frames. Rotating to the right or clicking will display the next frame from the file. Rotating to the left will display the previous frame. This enables some creative uses of image by rotating the wheel left and right. Note that the mechanical wheel pulse counter has not been designed yet.

Frame Pulse Counter

This value is the number of wheel pulses that will advance to the next frame when manual frame advance is set. Note that this disables using the rotary dial to advance frames.

Start Delay

This value delays the image display after the start button is pressed. It can be set in 1/10 seconds.

Upside Down Image

When set, the image pixels will be reversed. This allows the LED bar to be held upside down without making the image upside down.

Walk Direction

This option changes the direction that the file is read. This allows the image to play forward or reverse. This allows the user to walk in the other direction or to simply reverse the image from left to right. The default is right to left, looking back at the camera. Right to left is usually best when recording a sequence of images into a macro. The images will then play back in the same order. If left to right is chosen then the images recorded will play back in the reverse order. This is particularly important when spelling words with the individual letter image files.

Play Mirror Image

This plays the image twice, once forward and then in reverse. This creates a mirror like effect.

Scale Height to Fit

If the image is taller than 144/288 this option will ignore some pixels so that only 144/288 are actually displayed. It doesn't do any fancy scaling like Photoshop by looking at adjacent pixels, it simply ignores some of them so the image will fit.

144 to 288 Pixels (only when 2 strips)

This doubles every pixel so a 144 pixel image will fill all 288 pixels. Of course there isn't any increase in resolution, but it does fill two strips if you don't have a 288 pixel image.

Repeat Settings

This submenu contains settings that are used to repeat images. Except for the chaining option it also applies to built-in images.

Repeat Count

This is the number of times to repeat an image.

Repeat Delay

When repeat count is greater than one this value will cause a delay before the image starts again. It can be in 1/10's seconds.

Chain Files

When this option is on all the files from the current one to the end of the last file in the current folder will be displayed. Since filenames are always sorted alphabetically this option will show a sorted list of files. If some other order is desired it is also possible to use the macro command to record the file order for display.

Chain Repeats (only when chain files on)

The entire chain will be repeated this many times.

LED Strip Settings

This submenu has settings for the LED strip.

Strip Brightness

This is the brightness from 1 to 255. It was originally from 1 to 100%, but it was found that this didn't give very fine resolution of brightness on the low end, so it was decided to use the actual hardware value which has 255 level, not 100.

LED Strips

This is set to "one" or "two" depending on whether you have the second strip installed. The designer didn't figure out any way to automatically detect the second strip.

Gamma Correction

This option modifies the displayed colors at different brightness levels to more closely match our eyes color perception.

White Balance R

This manually sets the strip Red brightness.

White Balance G

This manually sets the strip Green brightness.

White Balance B

This manually sets the strip Blue brightness.

Show White Balance

Selecting this displays the before and after result of setting the RGB levels.

IPC File Operations

IPC files are files that can load settings and display images. They are text and can be modified by using any text editor on them. The extension is always IPC (Image Painter Command). There are menu items to create, run and delete them. Depending on the name some of them are automatically run under certain conditions as documented below.

IPC file only control the settings for files, none of the built-in image settings are saved in these files.

Save start.ipc

This creates a file with the current settings. It is run whenever a folder is opened.

Load start.ipc

This will manually run the file in the current folder.

Erase start.ipc

This does just what it says: removes the start.ipc file from the current folder.

Associated Files

These are command files just like the start.ipc file above, but they have the same name as a BMP file, but with the .ipc extension of course. The file is run when a BMP file is started for display. This allows specific settings to always be applied to a specified BMP file.

Save <filename>.ipc

This saves the current settings for the currently selected BMP file.

Load <filename>.ipc

This loads the settings for the currently selected BMP file. It can be used to verify that the settings are in fact correct.

Erase <filename>.ipc

This command deletes the file associated the currently selected BMP file.

Macros

Macro files are a powerful feature that can be used to record and run a set of images. There are 10 macro files allowed. They are named from 0.ipc to 9.ipc on the SD card in the root folder. They can also be edited on a computer by using a text editor if necessary. The easiest way to create one is to use the record menu entry as shown below.

Macro #(0-9)

This selects the current macro number. It shows a list of all the macros along with information on whether it already exists or not. Clicking on a line makes that macro # the active current macro # which is used by most of the following menu entries.

Run

This will run the currently selected macro number.

Repeat Count

This value repeats the macro run.

Repeat Delay (S)

There can be a delay between each run of the macro when repeat is on.

Recording

When set to true the settings and displayed images will be appended to the currently selected macro number. It should be turned off when all the actions have been recorded. Is the delete command if a sequence is to be started without appending.

Load

This loads the settings in a macro file. It is best used in combination with the save command.

Save

This saves the current settings in a macro file.

Delete

This erases the currently selected macro file. It is useful when a new sequence is to be recorded since the default record action is to append new actions.

Saved Settings

These settings are stored in the EEPROM section of the ESP32 and is a useful way to set all of the defaults during boot time.

There is a version number stored in EEPROM that will be compared when loading. I mismatch will cause an error message to appear. Saving the settings will clear this error.

Autoload Saved

This flag causes the EEPROM values to be loaded at boot time. If off then the factory settings will be loaded instead. If factory settings are desired the button should be pressed during booting. This will ignore the setting of this flag.

There is some quirkiness in the code that will not allow this flag to be turned off until the settings have been saved at least once. This only happens the first time the software is loaded on a new ESP32 board. Using the “save current settings” command will fix this forever.

Save Current Settings

This saves the current settings. Built-in options settings are also saved. Note that IPC files do not save any of the built-in settings.

Load Saved Settings

This manually loads the settings from EEPROM.

System Settings

Various system settings can be changed from this submenu. These include the display and various operations.

Display Brightness

This is the TFT display brightness from 1 to 100%. It is often desirable to lower it when working in strong darkness.

Display (Normal or Reverse)

This option reverses the colors on the display.

Menu Wrap

When this is off scrolling of both files and menu items is pinned at the top and bottom of the list. If wrap is on then rotation the dial to the right when at the last entry will go to the first entry and rotation left from the first entry to move to the last entry.

Show More Files

In run mode the top line always shows the file to be displayed. When this flag is on the following lines on the display will show the next few files in the current folder.

Show Folder

The top line in run mode shows the current file to be run. If this flag is on it will include the folder path. It should be turned off if the folder path is too long.

Progress Bar

When a file is running the top of the OLED display can show a progress bar. This option is used to turn it on or off.

Dial (Normal or Reverse)

This sets the direction of the dial that moves forward or backward through choices.

Dial Sensitivity

It is sometimes desirable to reduce the sensitivity of the dial to rotation. Wearing gloves or shivering fingers is an example. This allows a counter to be set such that more than one rotation detent will be required to produce a dial motion. For example, setting this to 2 means that the dial has to be moved two clicks before it would change the selection. Values from 1 to 5 can be set.

Dial Speed

This value adjusts how fast the dial needs to be moved to work with the dial sensitivity setting. It is effectively the time between clicks. It can set from 100 to 1000 and is approximately a time value in mSeconds. If you have the sensitivity set to a number greater than 1 then you have to rotate the dial faster than this number allows in order to cause the selection to change.

BlueTooth Link (On/Off)

This enables the BlueTooth link to be used with the phone app for control of the Image Painter.

Light Bar

This selection turns the Image Painter into an adjust LED light source. All of the LED's are turned on. The menu can then be used to change the Hue, saturation, and brightness levels. The maximum brightness is limited by the strip brightness that is set in the strip menu. This may be changed in the future so it will be possible to increase the brightness beyond that setting. Anybody have any strong thoughts on that?

The default allows setting the brightness, hue and saturation, but it can also be changed to use RGB values. The default increment unit is 10 for each click of the dial but that can also be adjust for finer resolution.

A long press is used to turn the light bar off again.

Reboot

Just like unplugging the power and plugging it back in.

Built-In Patterns

There are a number of built-in pattern generators. Some of them have settings that can be adjusted from the entry that appears on the first menu when a particular built-in is selected. The available built-ins are listed below.

Barber Pole

This creates a rolling red, white, and blue pattern. It will create diagonal patterns. The speed is controllable by setting the frame hold time.

Beats

Colored bars moving up and down.

Bouncy Balls

This uses physics calculations to simulate some colored bouncing balls. The number of balls can be set up to 8, each will have its own color. The speed can be changed using the same built-in menu. The frame hold time is not used. On an image it makes some interesting decaying inverted second order polynomial traces. Sorry about the math terms! Play with it.

Checker Board

Alternating black and white boxes. The speed is adjustable using the built-in menu or the frame hold time.

Confetti

Interesting color dots that appear and then fade out.

Cylon Eye

This was inspired from the BattleStar Galactica movies, the evil Cylon eyes.

Juggle

This shows some color bars that move up and down and cross over each other.

Meteor

This looks kind of like a meteor moving from one end of the strip to the other.

One Dot

Actually, this is four dots, red, green, blue and white. They move one at a time from one end to the other. It is probably mostly useful for testing the wand. That's why it was written originally.

Rainbow

The color, number of repeats, and color cycling can be controlled. It is even possible to add random sparkles to add a bit more interest. Try moving the wand in an arc for interesting rainbow-like patterns.

Random Bars

This produces solid colors that alternate randomly. An option sets black in between each color bar. Waving the wand can make some interesting patterns in the air.

Rainbow Pulse

This is rainbow that starts from the bottom, sweeps up, and then back down again until it is gone.

Sine Trails

A little more math here traces a color blob using the sine function.

Solid Color

This is probably one of the more useful patterns. It allows the LED strip to be used as a light source. The color can be controlled with the dial as well as the saturation and brightness. A setting allows the RGB values to be entered if that is desired. The brightness value is limited by the LED strip brightness setting from the main LED settings menu. So if more brightness is needed that value must be changed. The LED strip will stay on until a long press turns it off.

Twinkle

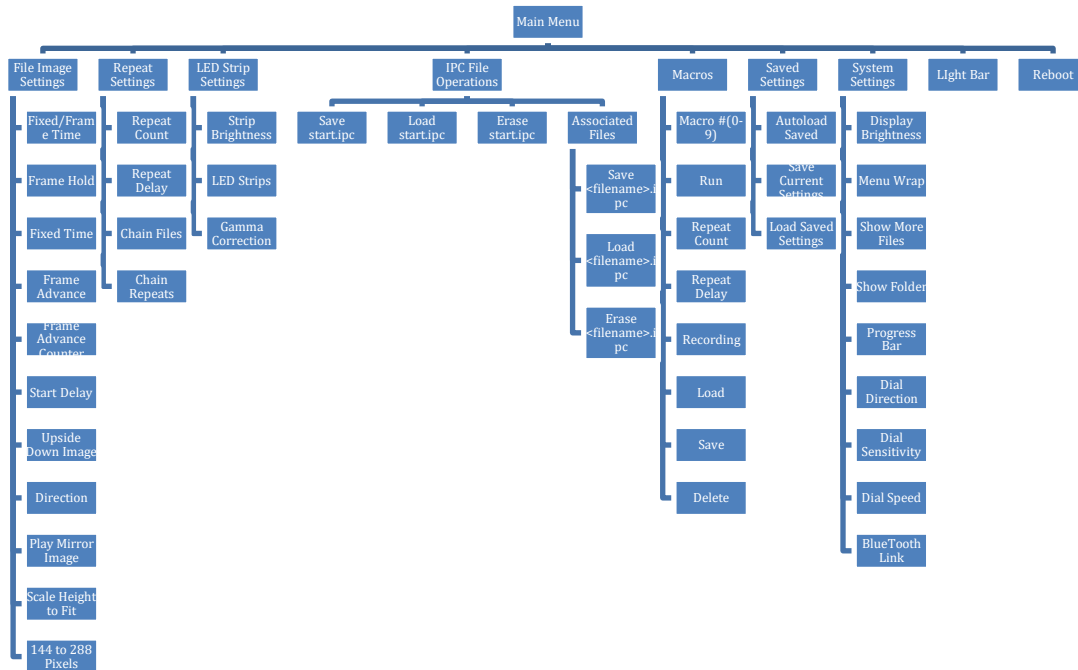
A display of dots of various colors that change color randomly.

Two Dots

This is like the One Dot, except that two dots are on the strip. They start at opposite ends and cross over in the middle. It was also written originally to test the code and the LED strip.

Menu structure

Here is a chart of the menus. Note that menus that are not applicable will not be displayed. For example, choosing Frame Time means that Fixed Time will not appear in the menu while choosing Fixed Time means that Frame Hold won't appear.



Construction

Yes, there are a few parts and a fair amount of wiring and soldering to do. If you aren't comfortable soldering, find someone who is! The wiring harnesses are probably the most work to make, but they are essential in order to hook everything up. I will include a photo of the ones I make that should make it easier to duplicate. Looking at the schematic will help understanding what is connected to what. The 3D printed parts can be purchased from me if you can't find anybody else to make them from the STL files. I print currently using PLA, so please don't leave them in the sun, they will melt and deform if it gets too hot!

All of the screws are metric and should be available from a well-stocked hardware store. I got mine from Ace Hardware.

All of the electronic components and the aluminum channel are available from Amazon. They are likely also available from other vendors. I bought the aluminum shiny ones because I didn't

find the black anodized ones until later. Some people worry that the shiny aluminum will show in the final image. In practice I haven't seen this, but I do plan to start using the black one after I finish using all the shiny ones I have in a box! I got the WS2812B LED strips from BangGood.com in China. Aliexpress.com also carries the WD2812B. Be aware that the shipping time from China can be quite long. Amazon is faster.

I am considering making a parts bag available with all the hardware parts needed. I buy them in quantity so it would be easy enough to put a bag together.

You can solder up a bunch of wires and connectors to connect everything inside the cpu/display box like I did for my first prototype, but I have designed a printed circuit holding all the parts and I can sell those for \$10+shipping. It does make the assembly much less painful. I can also make the PCB version available with the ESP32 already soldered and programmed.

Parts List

Electronic Parts

1. Heltec ESP32 Devkit32.
https://www.amazon.com/gp/product/B07DKD79Y9/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
2. Micro SD card reader.
https://www.amazon.com/gp/product/B07BJ2P6X6/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
3. Rotating switch.
https://www.amazon.com/gp/product/B07DM2YMT4/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
4. USB rechargeable battery.
https://www.amazon.com/gp/product/B07MCDT8FW/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1
5. USB connector.
https://www.amazon.com/gp/product/B072DWP59M/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
6. Logic level shifter.
https://www.amazon.com/gp/product/B07LG646VS/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
7. 5 pin, 4 pin, and 3 pin connectors.
https://www.amazon.com/gp/product/B07GGPQXFC/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
https://www.amazon.com/gp/product/B075K3M1TB/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
https://www.amazon.com/gp/product/B075K6N7DF/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

8. LED channel.
https://www.amazon.com/gp/product/B01LL2SLME/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
9. WS2812B LED strips.
https://www.amazon.com/gp/product/B07BKQZLDL/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

3D Printed Parts

1. Battery holder.
2. Rotating handle base.
3. Rotating handle.
4. Electronics box. There are two of these, one for the PCB, and one for discrete parts.
5. Handle.
6. LED strip end protectors.

Software and Libraries

I will make a compiled binary available on the github entry for this project if you don't want to collect all these files and compile the code. It can be built with the Arduino IDE, but I personally prefer Microsoft Visual Studio Community version along with the Visual Micro extension for Arduino, which is a bargain at \$19/year.

<https://github.com/MartinNohr/LedImagePainter>

1. Arduino IDE, Visual Studio Code, or Microsoft Visual Studio with Visual Micro or any other development system that can compile the C and C++ code for the ESP32.
2. ESP32 support libraries for Heltec wifi devkit32
3. ArduinoJSON library (the one by Benoit Blanchon)
4. FastLED library
5. Ringbufcpp library (this is not used in the latest version of the software)
6. Stackarray library

Wiring Harnesses

These are somewhat time consuming to make. Soldering skills are required. See pictures for what they look like.

1. Internal, for SD card reader, level shifter, LED/Power, and rotating button.
2. Battery and LED's.
3. Rotating button.