



Video Playing 2.1" Round Ornament TFT

Created by Ruiz Brothers



<https://learn.adafruit.com/2-1-round-ornament-tft>

Last updated on 2024-06-03 03:57:10 PM EDT

Table of Contents

Overview	3
<hr/>	
<ul style="list-style-type: none">• Parts• Prerequisite Guides	
Circuit Diagram	6
<hr/>	
<ul style="list-style-type: none">• Adafruit Library for Fritzing• Wired Connections	
Converting Videos	7
<hr/>	
<ul style="list-style-type: none">• Video File Preparation• Online Converter• Format SD• Create VIDEOS folder• Conversion Completed	
3D Printing	10
<hr/>	
<ul style="list-style-type: none">• 3D Printed Parts• Slice with Settings for PLA material• Design Source Files	
Software Setup and Use	11
<hr/>	
<ul style="list-style-type: none">• Prep SD Card• Upload UF2 File• Advanced Users: Source Code in Arduino IDE	
Assemble	18
<hr/>	
<ul style="list-style-type: none">• Prep SD breakout• Mount display• Mounting The Frame• Mount SD breakout• Connect SD cable• Display Ribbon• Attach Lid	

Overview



Build an Ornament that plays Holiday video clips using an Adafruit Qualia ESP32-S3 and a 2.1" round TFT display.

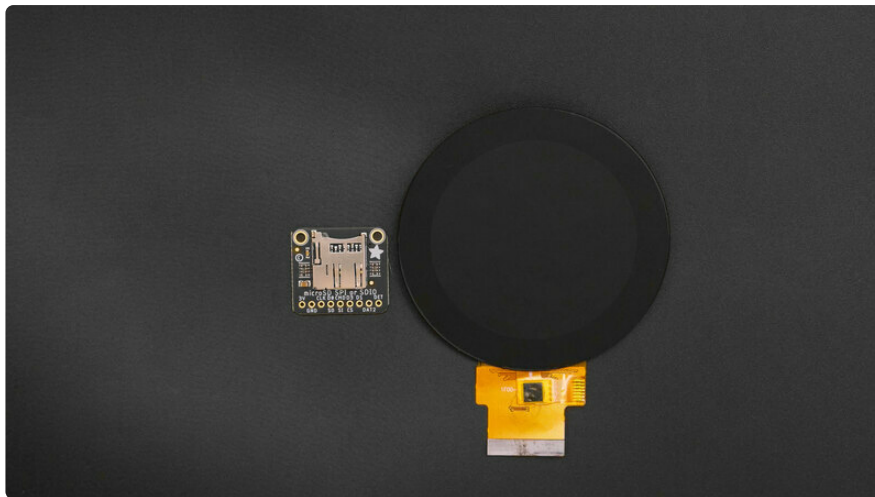


Load a series of your favorite video files on a micro SD card and display them on the round TFT LCD screen. Cycle through them by touching the screen.

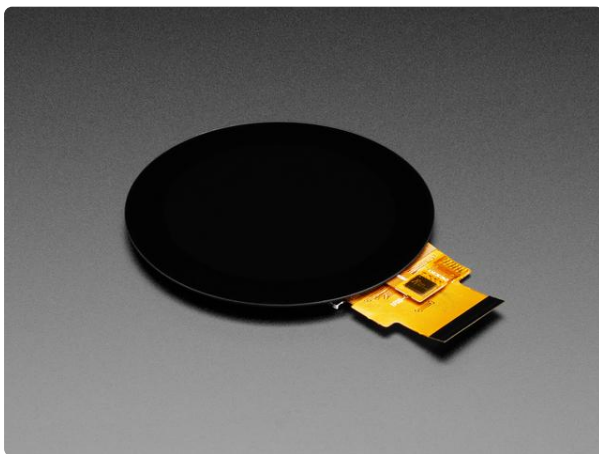


3D print the parts to secure the display, Qualia ESP32-S2, and micro SD breakout.

The case uses ornament hooks so you can decorate your tree for the season!



Parts



Round RGB 666 TTL TFT Display - 2.1" 480x480 - Capacitive Touch

This is a screen for advanced hackers who like the look of a nice, round TFT screen with tons of pixels. The 2.1" diagonal sized display has 480x480 16-bit full-color pixels and...

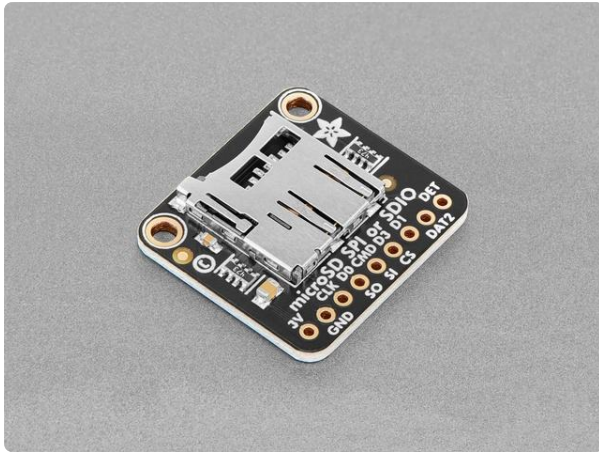
<https://www.adafruit.com/product/5792>



Adafruit Qualia ESP32-S3 for TTL RGB-666 Displays

There's a few things everyone loves: ice cream, kittens, and honkin' large TFT screens. We're no strangers to small TFT's -

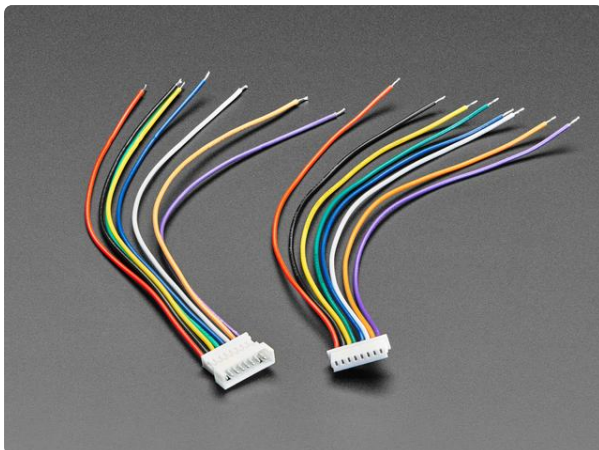
<https://www.adafruit.com/product/5800>



Adafruit Micro SD SPI or SDIO Card Breakout Board - 3V ONLY!

Micro SD cards and microcontrollers go together like micro-peanutbutter-and-jelly: SD cards are inexpensive, durable, easy to find at any shop, come in many sizes and can plug into any...

<https://www.adafruit.com/product/4682>



1.25mm Pitch 8-pin Cable Matching Pair - 10 cm long

These solid 1.25mm pitch connector cable pairs are great when you need something that can carry a couple of amps of current and are easy to connect and disconnect. These cables...

<https://www.adafruit.com/product/4976>



SD/MicroSD Memory Card (8 GB SDHC)

Add mega-storage in a jiffy using this 8 GB class 4 micro-SD card. It comes with a SD adapter so you can use it with any of our shields or adapters. Preformatted to FAT so it works out...

<https://www.adafruit.com/product/1294>



Pink and Purple Woven USB A to USB C Cable - 1 meter long

This cable is not only super-fashionable, with a woven pink and purple Blinka-like pattern, it's also made for USB C for our modernized breakout boards, Feathers, and...

<https://www.adafruit.com/product/5153>

M2.5x5mm Screws

6 x M2.5x5mm Screws

M2.5x5mm Screws

Prerequisite Guides

Take a moment to review the following guides to learn more about the products.

- [Adafruit Qualia ESP32-S3 Guide \(https://adafru.it/19ak\)](https://adafru.it/19ak)

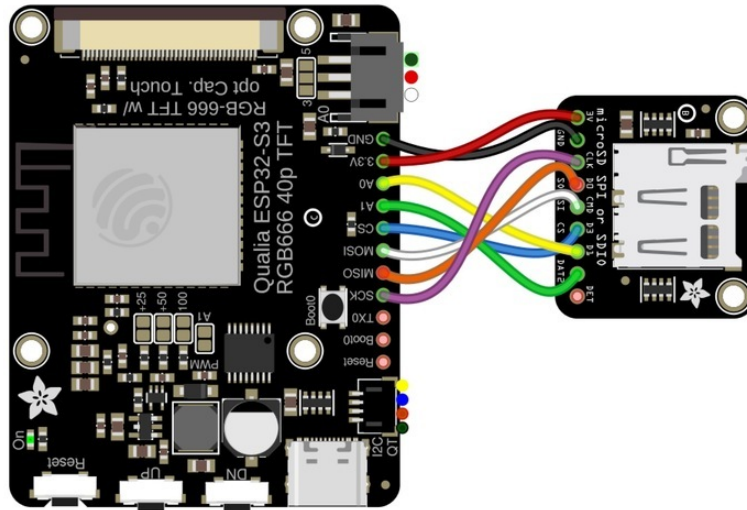


Circuit Diagram

The diagram below provides a general visual reference for wiring of the components once you get to the **Assembly** page. This diagram was created using the software package [Fritzing \(https://adafru.it/oEP\)](https://adafru.it/oEP).

Adafruit Library for Fritzing

Adafruit uses the Adafruit's Fritzing parts library to create circuit diagrams for projects. You can download the library or just grab individual parts. Get the library and parts from [GitHub - Adafruit Fritzing Parts \(https://adafru.it/AYZ\)](https://adafru.it/AYZ).



Wired Connections

The Qualia ESP32-S3 is powered by a 5V 1A USB power supply.

- **SCK** from Qualia S3 to **CLK** on Micro SD
- **MISO** from Qualia S3 to **SO** on Micro SD
- **MOSI** from Qualia S3 to **SI** on Micro SD
- **CS** from Qualia S3 to **CS** on Micro SD
- **A1** from Qualia S3 to **DAT2** on Micro SD
- **A0** from Qualia S3 to **D1** on Micro SD
- **3.3V** from Qualia S3 to **3V** on Micro SD
- **GND** from Qualia to **GND** on Micro SD

Converting Videos

Video File Preparation

You'll want to use a video file in a common format such as H.264 (.MP4 or .MOV) with a duration of anywhere from 10 seconds to 3 minutes in length. Your video clip should ideally have a 1:1 aspect ratio of 480x480.

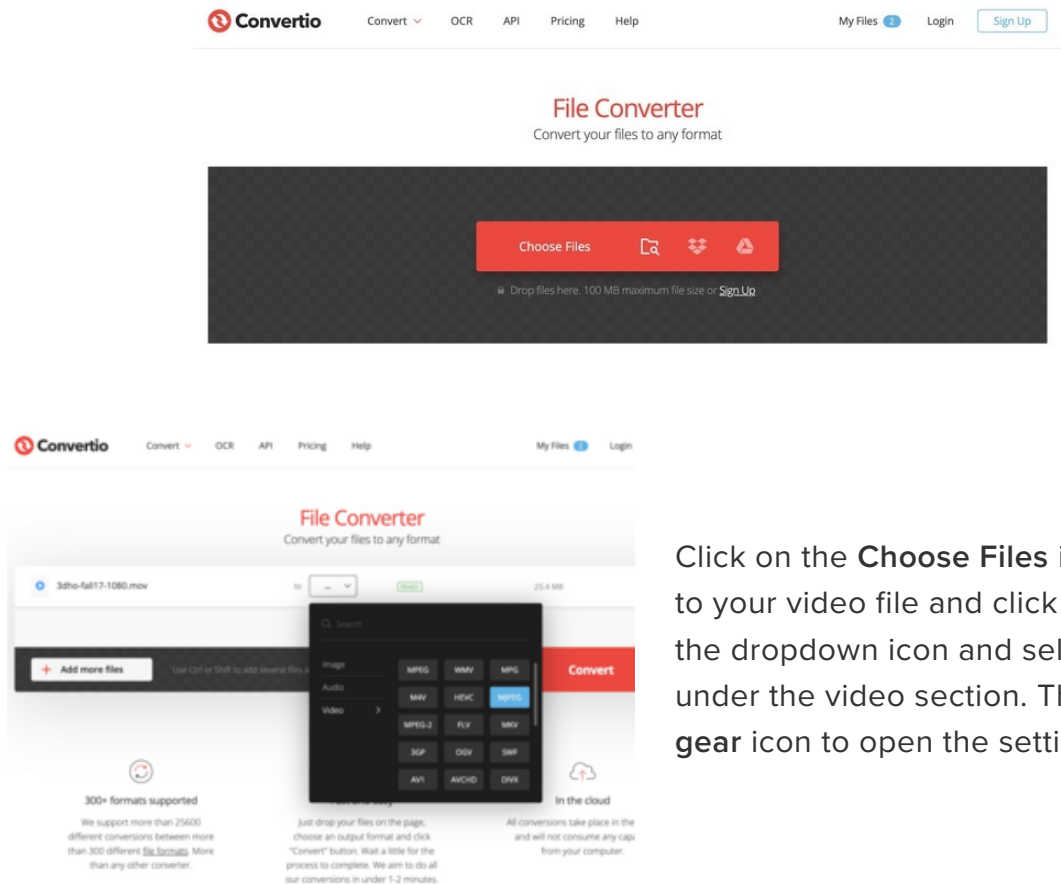
The 2.1" display has a screen resolution of 480x480.

Online Converter

You can use the website linked below to convert your video file into MJPEG.
(maximum file size is 100MB.)

Convert IO Website

<https://adafru.it/19aZ>



Click on the **Choose Files** icon. Navigate to your video file and click **Open**. Click on the dropdown icon and select **MJPEG** under the video section. Then, click on the gear icon to open the settings dialog.

Use the following settings for best playback performance.

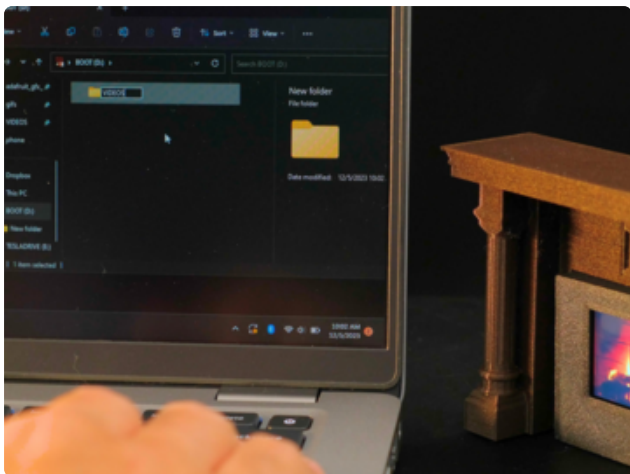
- Codec: MJPEG
- Quality: Highest
- Resize: Custom / 480 x 480
- Resize Method: Zoom and crop
- Frame rate : 10 FPS
- Rotate: Rotate by 90 degrees clockwise (this is to compensate for the screen orientation when mounted)

Click on the Convert button when finished.



Format SD

Format a micro SD card to FAT32.



Create VIDEOS folder

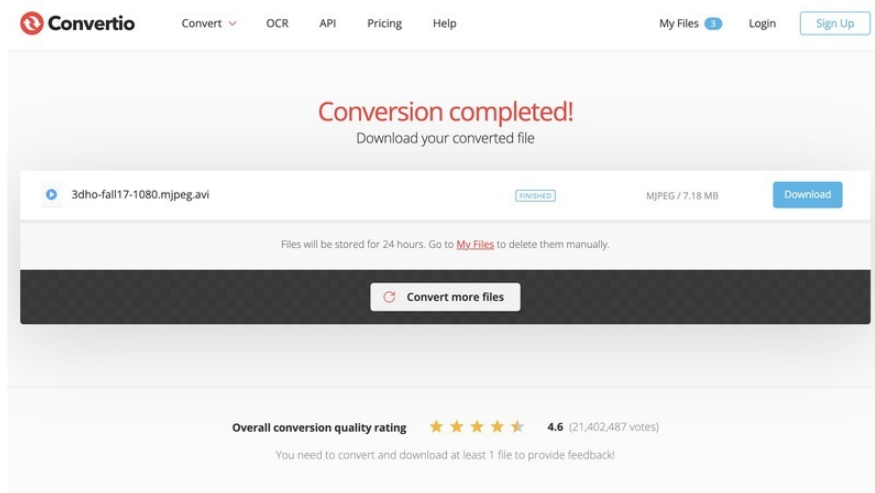
Make a new folder in the SD card and name it **VIDEOS**.

Conversion Completed

Click the **Download** button when the upload and conversion is complete.

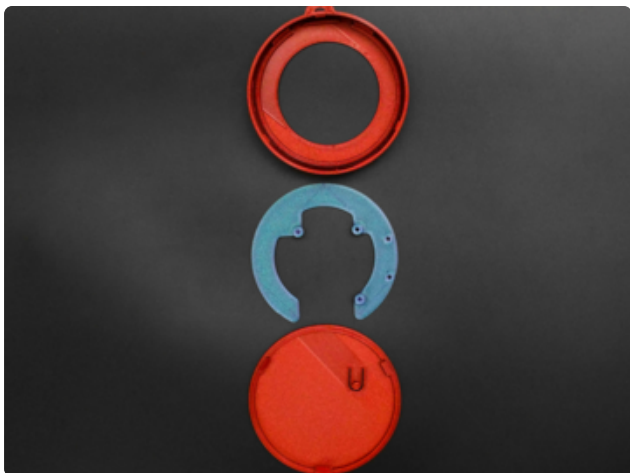
Use the [VLC media player app \(https://adafru.it/19b0\)](https://adafru.it/19b0) to playback the video for reviewing.

Rename the video file so it only contains the **.MJPEG** extension. Then, drag and drop it into the **VIDEOS** folder on the micro SD card.



Mac users may need to delete .DS files inside the VIDEOS folder

3D Printing



3D Printed Parts

STL files for 3D printing are oriented to print "as-is" on FDM style machines. Parts are designed to 3D print without any support material using PLA filament. Original design source may be downloaded using the links below.

Edit Design

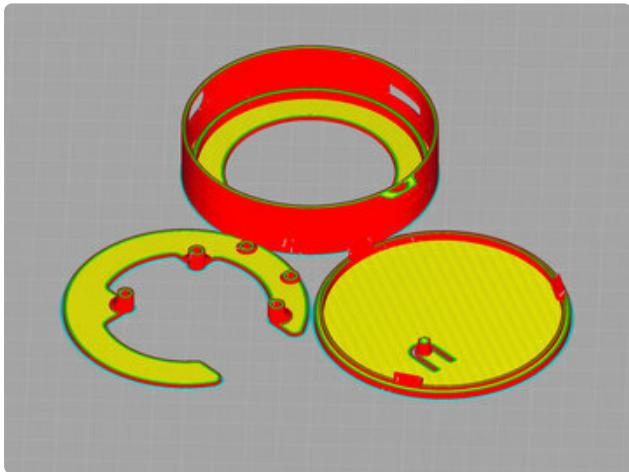
<https://adafru.it/19bi>

Download STLs

<https://adafru.it/19bk>

2.1_Round_Ornament_STEP.zip

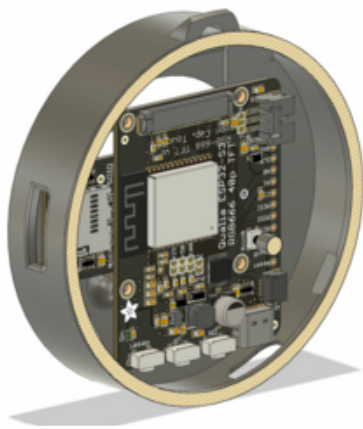
<https://adafru.it/19bl>



Slice with Settings for PLA material

The parts were sliced using CURA using the slice settings below.

PLA filament 200c extruder
0.2 layer height
10% gyroid infill
60mm/s print speed
60c heated bed



Design Source Files

The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, STL and more. Electronic components like Adafruit's boards, displays, connectors and more can be downloaded from the [Adafruit CAD parts GitHub Repo \(https://adafru.it/RvF\)](https://adafru.it/RvF).

Software Setup and Use



Prep SD Card

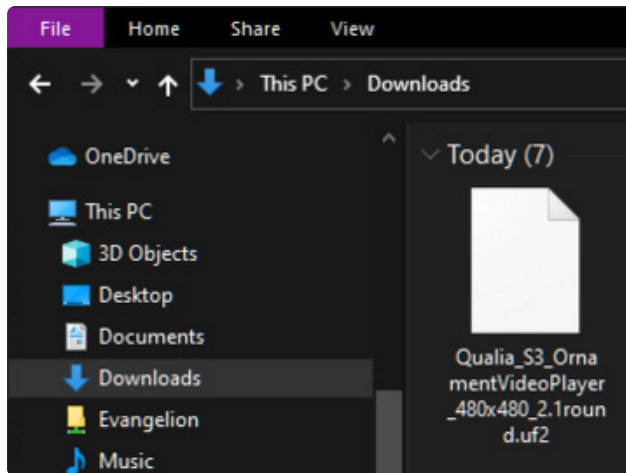
Create a folder called **VIDEOS** on your SD card. Save your converted video files in this folder. Eject the SD card from your computer and insert it into the microSD card breakout.

Upload UF2 File

The round ornament code is available as a pre-compiled .UF2 file for the 2.1" 480x480 display that you can drag and drop onto your Qualia S3 board.

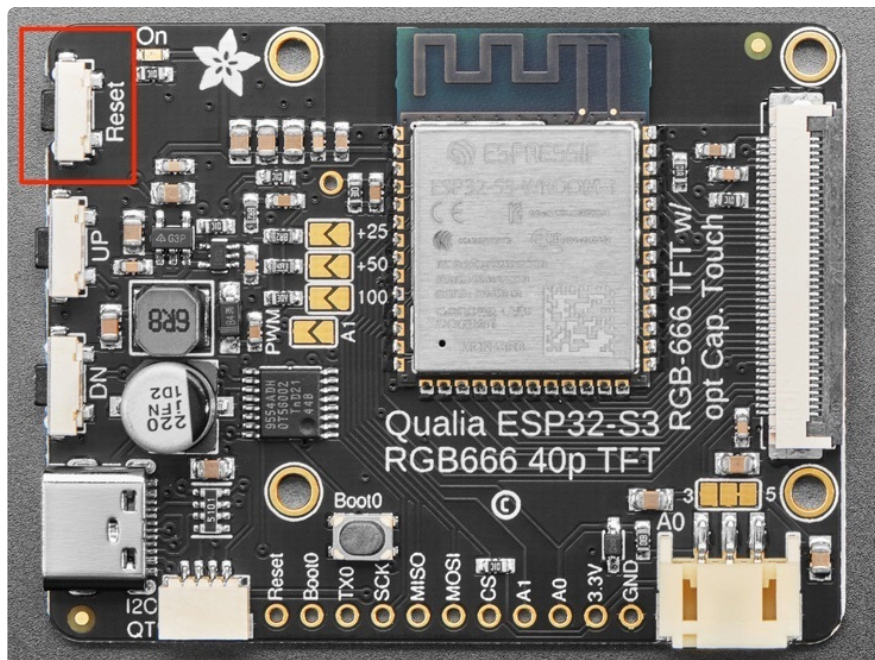
Qualia S3 Ornament UF2 File

<https://adafru.it/19bn>



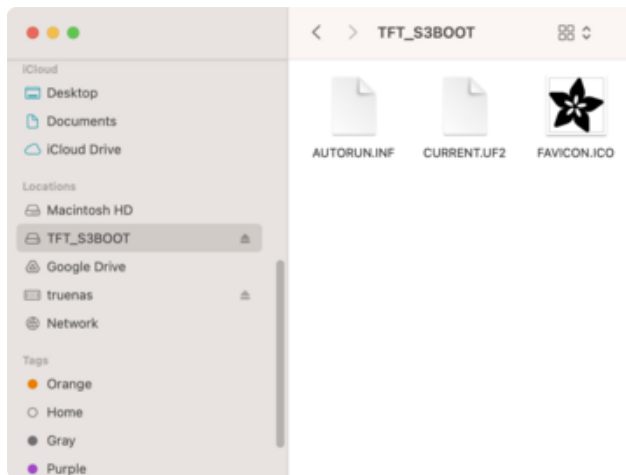
Click the link above to download the UF2 file.

Save it wherever is convenient for you.



Plug your board into your computer, using a known-good data-sync USB cable, directly, or via an adapter if needed.

Double-click the **reset** button (highlighted in red above), wait about a half a second and then tap reset again.



You will see a new disk drive appear called **TFT_S3BOOT**.

Drag the **Qualia_S3_OrnamentVideoPlayer_480x480_2.1rou** file to **TFT_S3BOOT**.

The code will begin running by playing the first video file on the SD card. Touch the center of the screen to advance to the next video file.

Advanced Users: Source Code in Arduino IDE

Only attempt this if you are comfortable with Git and advanced Arduino use. Otherwise, use the precompiled UF2.

Qualia S3 Round Ornament Arduino Code

<https://adafru.it/19bo>

```
// SPDX-FileCopyrightText: 2023 Limor Fried for Adafruit Industries
//
// SPDX-License-Identifier: MIT

/*****
 * Motion JPEG Image Viewer
 * This is a simple Motion JPEG image viewer example
 *****/

encode with
ffmpeg -i "wash.mp4" -vf
"fps=10,vflip,hflip,scale=-1:480:flags=lanzcros,crop=480:480" -pix_fmt yuvj420p -q:v
9 wash.mjpeg

*****/
#define MJPEG_FOLDER      "/videos" // cannot be root!
#define MJPEG_OUTPUT_SIZE (480 * 480 * 2) // memory for a output image frame
#define MJPEG_BUFFER_SIZE (MJPEG_OUTPUT_SIZE / 5) // memory for a single JPEG frame
#define MJPEG_LOOPS      0

#include <Arduino_GFX_Library.h>
#include <Adafruit_CST8XX.h>
// #include <SD.h> // uncomment either SD or SD_MMC
#include <SD_MMC.h>

Arduino_XCA9554SWSPI *expander = new Arduino_XCA9554SWSPI(
  PCA_TFT_RESET, PCA_TFT_CS, PCA_TFT_SCK, PCA_TFT_MOSI,
  &Wire, 0x3F);
```



```

Arduino_ESP32RGBPanel *rgbpanel = new Arduino_ESP32RGBPanel(
    TFT_DE, TFT_VSYNC, TFT_HSYNC, TFT_PCLK,
    TFT_R1, TFT_R2, TFT_R3, TFT_R4, TFT_R5,
    TFT_G0, TFT_G1, TFT_G2, TFT_G3, TFT_G4, TFT_G5,
    TFT_B1, TFT_B2, TFT_B3, TFT_B4, TFT_B5,
    1 /* hsync_polarity */, 50 /* hsync_front_porch */, 2 /* hsync_pulse_width */,
44 /* hsync_back_porch */,
    1 /* vsync_polarity */, 16 /* vsync_front_porch */, 2 /* vsync_pulse_width */,
18 /* vsync_back_porch */
    //,1, 30000000
);

Arduino_RGB_Display *gfx = new Arduino_RGB_Display(
// 2.1" 480x480 round display
    480 /* width */, 480 /* height */, rgbpanel, 0 /* rotation */, true /*
auto_flush */,
    expander, GFX_NOT_DEFINED /* RST */, TL021WVC02_init_operations,
sizeof(TL021WVC02_init_operations));

Adafruit_CST8XX ctp = Adafruit_CST8XX(); // This library also supports FT6336U!
#define I2C_TOUCH_ADDR 0x15
bool touchOK = false;

#include <SD_MMC.h>

#include "MjpegClass.h"
static MjpegClass mjpeg;
File mjpegFile, video_dir;
uint8_t *mjpeg_buf;
uint16_t *output_buf;

unsigned long total_show_video = 0;

void setup()
{
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    //while(!Serial) delay(10);
    Serial.println("MJPEG Video Playback Demo");

#ifdef GFX_EXTRA_PRE_INIT
    GFX_EXTRA_PRE_INIT();
#endif

    // Init Display
    Wire.setClock(400000); // speed up I2C
    if (!gfx->begin()) {
        Serial.println("gfx->begin() failed!");
    }
    gfx->fillScreen(BLUE);

    expander->pinMode(PCA_TFT_BACKLIGHT, OUTPUT);
    expander->digitalWrite(PCA_TFT_BACKLIGHT, HIGH);

    //while (!SD.begin(ss, SPI, 64000000UL))
    //SD_MMC.setPins(SCK /* CLK */, MOSI /* CMD/MOSI */, MISO /* D0/MISO */);
    SD_MMC.setPins(SCK, MOSI /* CMD/MOSI */, MISO /* D0/MISO */, A0 /* D1 */, A1 /*
D2 */, SS /* D3/CS */); // quad MMC!
    while (!SD_MMC.begin("/root", true))
    {
        Serial.println(F("ERROR: File System Mount Failed!"));
        gfx->println(F("ERROR: File System Mount Failed!"));
        delay(1000);
    }
    Serial.println("Found SD Card");

    // open filesystem
    //video_dir = SD.open(MJPEG_FOLDER);
    video_dir = SD_MMC.open(MJPEG_FOLDER);

```

```

if (!video_dir || !video_dir.isDirectory()){
    Serial.println("Failed to open " MJPEG_FOLDER " directory");
    while (1) delay(100);
}
Serial.println("Opened Dir");

mjpeg_buf = (uint8_t *)malloc(MJPEG_BUFFER_SIZE);
if (!mjpeg_buf) {
    Serial.println(F("mjpeg_buf malloc failed!"));
    while (1) delay(100);
}
Serial.println("Allocated decoding buffer");

output_buf = (uint16_t *)heap_caps_aligned_alloc(16, MJPEG_OUTPUT_SIZE,
MALLOC_CAP_8BIT);
if (!output_buf) {
    Serial.println(F("output_buf malloc failed!"));
    while (1) delay(100);
}

expander->pinMode(PCA_BUTTON_UP, INPUT);
expander->pinMode(PCA_BUTTON_DOWN, INPUT);

if (!ctp.begin(&Wire, I2C_TOUCH_ADDR)) {
    Serial.println("No touchscreen found");
    touchOK = false;
} else {
    Serial.println("Touchscreen found");
    touchOK = true;
}
}

void loop()
{
    /* variables */
    int total_frames = 0;
    unsigned long total_read_video = 0;
    unsigned long total_decode_video = 0;
    unsigned long start_ms, curr_ms;
    uint8_t check_UI_count = 0;
    int16_t x = -1, y = -1, w = -1, h = -1;
    total_show_video = 0;

    if (mjpegFile) mjpegFile.close();
    Serial.println("looking for a file...");

    if (!video_dir || !video_dir.isDirectory()){
        Serial.println("Failed to open " MJPEG_FOLDER " directory");
        while (1) delay(100);
    }

    // look for first mjpeg file
    while ((mjpegFile = video_dir.openNextFile()) != 0) {
        if (!mjpegFile.isDirectory()) {
            Serial.print(" FILE: ");
            Serial.print(mjpegFile.name());
            Serial.print(" SIZE: ");
            Serial.println(mjpegFile.size());
            if ((strstr(mjpegFile.name(), ".mjpeg") != 0) || (strstr(mjpegFile.name(),
".MJPEG") != 0)) {
                Serial.println(" <---- found a video!");
                break;
            }
        }
        if (mjpegFile) mjpegFile.close();
    }

    if (!mjpegFile || mjpegFile.isDirectory())
    {

```

```

    Serial.println(F("ERROR: Failed to find a MJPEG file for reading,
resetting..."));
    //gfx->println(F("ERROR: Failed to find a MJPEG file for reading"));

    // We kept getting hard crashes when trying to rewindDirectory or close/open dir
    // so we're just going to do a softreset
    esp_sleep_enable_timer_wakeup(1000);
    esp_deep_sleep_start();
}

bool done_looping = false;
while (!done_looping) {
    mjpegFile.seek(0);
    total_frames = 0;
    total_read_video = 0;
    total_decode_video = 0;
    total_show_video = 0;

    Serial.println(F("MJPEG start"));

    start_ms = millis();
    curr_ms = millis();
    if (! mjpeg.setup(&mjpegFile, mjpeg_buf, output_buf, MJPEG_OUTPUT_SIZE, true /*
useBigEndian */)) {
        Serial.println("mjpeg.setup() failed");
        while (1) delay(100);
    }

    while (mjpegFile.available() && mjpeg.readMjpegBuf())
    {
        // Read video
        total_read_video += millis() - curr_ms;
        curr_ms = millis();

        // Play video
        mjpeg.decodeJpg();
        total_decode_video += millis() - curr_ms;
        curr_ms = millis();

        if (x == -1) {
            w = mjpeg.getWidth();
            h = mjpeg.getHeight();
            x = (w > gfx->width()) ? 0 : ((gfx->width() - w) / 2);
            y = (h > gfx->height()) ? 0 : ((gfx->height() - h) / 2);
        }
        gfx->draw16bitBeRGBBitmap(x, y, output_buf, w, h);
        total_show_video += millis() - curr_ms;

        curr_ms = millis();
        total_frames++;
        check_UI_count++;
        if (check_UI_count >= 5) {
            check_UI_count = 0;
            Serial.print('.');

            if (! expander->digitalRead(PCA_BUTTON_DOWN)) {
                Serial.println("\nDown pressed");
                done_looping = true;
                while (! expander->digitalRead(PCA_BUTTON_DOWN)) delay(10);
                break;
            }
            if (! expander->digitalRead(PCA_BUTTON_UP)) {
                Serial.println("\nUp pressed");
                done_looping = true;
                while (! expander->digitalRead(PCA_BUTTON_UP)) delay(10);
                break;
            }
        }

        if (touchOK && ctp.touched()) {

```

```

        CST_TS_Point p = ctp.getPoint(0);
        Serial.printf("(%d, %d)\n", p.x, p.y);
        done_looping = true;
        break;
    }
}
}
int time_used = millis() - start_ms;
Serial.println(F("MJPEG end"));

float fps = 1000.0 * total_frames / time_used;
total_decode_video -= total_show_video;
Serial.printf("Total frames: %d\n", total_frames);
Serial.printf("Time used: %d ms\n", time_used);
Serial.printf("Average FPS: %0.1f\n", fps);
Serial.printf("Read MJPEG: %lu ms (%0.1f %%) \n", total_read_video, 100.0 *
total_read_video / time_used);
Serial.printf("Decode video: %lu ms (%0.1f %%) \n", total_decode_video, 100.0 *
total_decode_video / time_used);
Serial.printf("Show video: %lu ms (%0.1f %%) \n", total_show_video, 100.0 *
total_show_video / time_used);
}
}

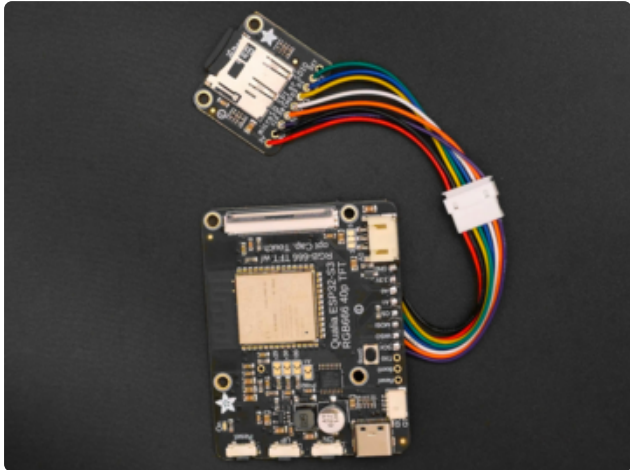
```

The source code for the round ornament is [available on GitHub \(https://adafru.it/19bo\)](https://adafru.it/19bo). It consists of an Arduino script .ino file and a header file. You will need both files to compile it in the Arduino IDE. There are a few items you'll need to manually configure in the Arduino IDE:

- The header file requires the ESP32_JPEG library, which isn't currently available in the Arduino IDE library bundle. You'll need to install it manually from its [GitHub repository \(https://adafru.it/19b7\)](https://adafru.it/19b7).
- Currently the Arduino GFX library is not compatible with the ESP BSP 3.0 since it uses IDF 5. You will need to use an older BSP package and manually add the Qualia S3 board to your local installation.

If you defeat these dragons though, you can update the code to run on different RGB-666 displays and customize any other parameters that you want.

Assemble



Prep SD breakout

Use an 8 pin matching cable pair to easily connect the SD breakout board to the pins on the Qualia board. Follow the wiring diagram [here \(https://adafru.it/19bp\)](https://adafru.it/19bp).



Mount display

Take note of how the ribbon cable attaches to the Qualia board. Align the display to the cutout and place face down between the walls inside the case.

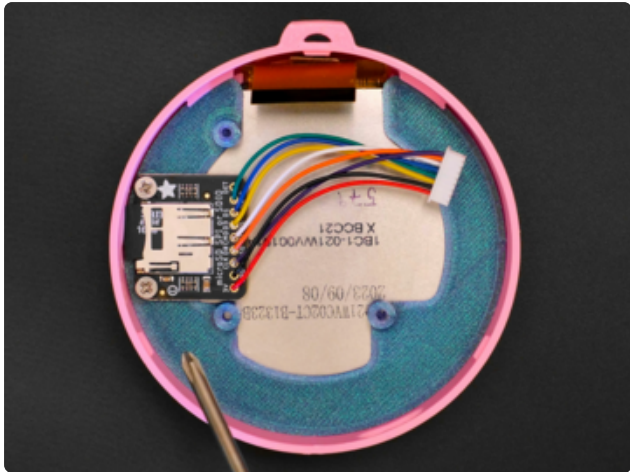
Slightly bend the case while gently pressing the edges of the display to fit.



Mounting The Frame

Align cutout on the mounting frame to the ribbon cable

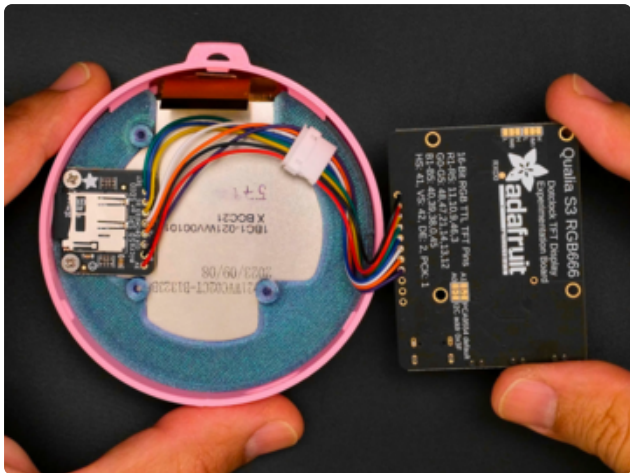
Slide the mounting frame into the case at an angle, between the snaps on the case.



Mount SD breakout

Align the SD breakout to the slot on the case.

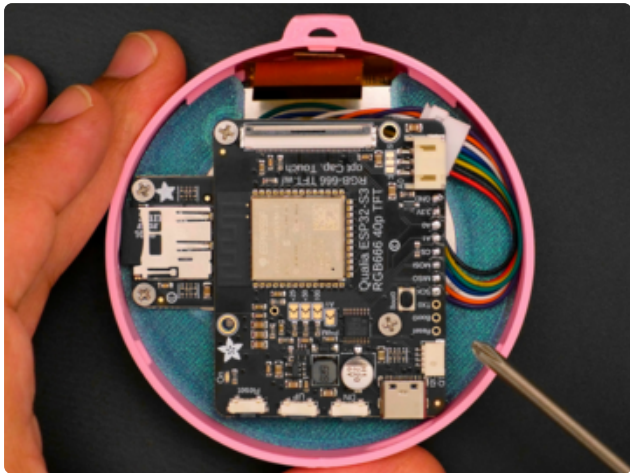
Use M2.5x5mm screws to mount the SD breakout board to the frame.

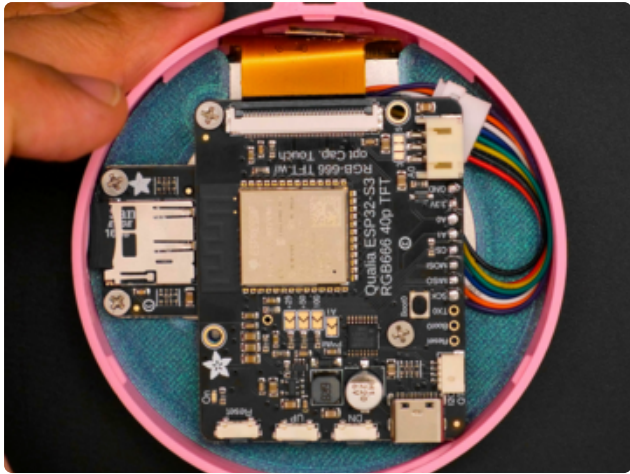


Connect SD cable

Connect the SD cables to the Qualia. Coil the cable and place the connector between the three taller standoffs.

Use M2.5x5mm screws to secure the Qualia to the frame.

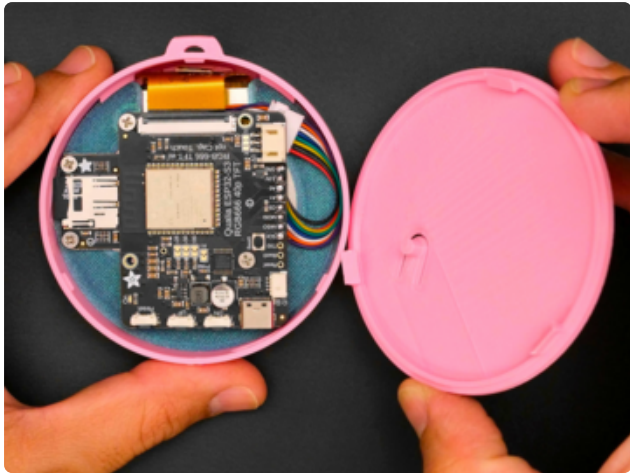




Display Ribbon

Carefully lift the connect bracket on the Qualia.

Place the display ribbon cable into the connector on the Qualia board and gently press the bracket back down to secure the ribbon cable into place.



Attach Lid

Align the snaps on the lid the nubs on the inside of the case to close the enclosure.



Plug in a USB battery pack or wall adapter and hang your video ornament!

