Table of Contents

Overview 3
• Countdown to fun
• Parts & Materials

Build the Circuit 6
• Crypto Countdown Case Circuit
• Switches
• Wire the Switches
• ON/OFF Circuit
• Display
• Power Button Lighting
• NeoPixel Indicator Lamp
• BEEEEEIP
• Wiring

Upload the Code 26

Make the Case 34
• Distressing
• Foam
• Mounting Panel
• Mounting the Parts
• Standoffs
• Power Button
• LED Indicator
• Piezo Mounting
• USB Connector
• Final Steps

Deploy the Case 63
• Usage
Picture this: You are a Cold War-era spymaster, stationed in West Germany. One of your operatives has risked life and limb to get her hands on this mysterious case. You inspect and then carefully open it to be greeted by an array of toggle switches, a countdown timer, and a power switch. Disarm it, and you'll have your hands on the vital launch code for a captured missile. Fail, and, well, it's best not to consider failure...

Your best cryptographers have intercepted the following transmission:

кра́сный - бе́лый - си́ний - зелёный - жёлтый

Thinking back to your language training at Langley, you translate this to:

RED - WHITE - BLUE - GREEN - YELLOW

That's it! These colors must be the secret to disarming the case!

You dare to turn on the power, the case arms itself, and a three minute countdown begins. You enter in the toggle switches, hopefully in the correct order.

The timer stops. Whew.

A secret code scrolls onto the display. The launch codes you needed! A job well done.
Countdown timers with switches, lights, and beeps are a staple of action thrillers. This is a fun and fairly simple prop to build using a briefcase or other similar container.

You could use it for a spy-themed escape room or mystery dinner, or as a prop for a movie or play. Better yet, use it to deliver a message to someone you know who likes solving puzzles.

You can easily customize the timer, switch combination, and secret message in the included Arduino sketch, or you can add different, complex behaviors, create your own Mastermind-like game, and more.
Parts & Materials

In addition to the parts listed on the right side of this guide, you'll want an interesting case in which to mount the electronics. I used this one which is similar to a Pelican case, albeit lower strength and lower cost. If you happen to have a Halliburton aluminum attaché case, that would be perfect!

If you'd like to follow this guide exactly, that one's a good choice, as I've included CAD files for panels that fit that case. You can laser cut and etch two sheets of 3mm acrylic to make the panel, or print the template and use it as a guide for hand cutting cardboard or thin plywood with a band saw, coping saw, or knife.

Additionally, you'll need some wire for making connections, a soldering iron and solder, and a small screwdriver.

To distress the case I used watered-down black acrylic paint, the kind you can get at a craft or hobby store. I also used gray spray primer and red glossy spray paint for the star. You'll want some cardboard, or better yet, Mylar stencil sheet to make the start stencil, and access to a laser cutter (or use a hobby knife) to cut the pattern.
Build the Circuit

Crypto Countdown Case Circuit

The case's circuit consists of:

- Inputs
  - on/off button
  - five toggle switches
• Outputs
  ◦ 14-segment alphanumeric LED display
  ◦ piezo buzzer
  ◦ NeoPixel RGB indicator LED

• Processing
  ◦ Feather M0 microcontroller

• Power
  ◦ 4400mAh Lithium Ion battery
  ◦ USB port for charging (and programming)

This diagram shows how you'll connect things, using the Terminal Block Breakout FeatherWing for wiring.
Switches

These toggle switches are not only really cool looking, but they are kind of fancy, too! They have some circuitry inside to allow their LEDs to light up, and can operate in one of three modes.

There are three connections to a switch: "GND", "+", and "=D" (also called "headlamp"). Here's how to use them with a Feather M0 (although this may be applied to other Feathers, Arduinos, or most any microcontroller):

To use a switch like a normal toggle

• connect a wire from Feather GND pin to switch Headlamp contact
• connect a wire from Feather Enable pin to switch + contact
• toggle the switch to from OFF to ON to close and open the circuit, sending the Enable pin LOW or HIGH to power the Feather off or on

To use a switch as a toggle, with the built-in switch LED permanently lit

• connect Feather GND pin to switch GND contact
• connect Feather 3v3 pin to switch Headlamp contact
• connect a Feather input pin, such as pin 11, to switch + contact

Note: This diagram does not show the small transistor switching circuit you'll build on the prototyping area of the Terminal Block FeatherWing -- see below for more info
toggle the switch from OFF to ON to send digital pin reading from HIGH to LOW

You will need to set the Feather pin mode to INPUT_PULLDOWN for this to work properly. This is possible on M0-based boards, for others you'll need to create a pulldown circuit with two resistors.

Here's a simple example sketch to use when testing the switch in permanently lit mode.

```cpp
//Illuminated toggle switch
//permanently lit mode
//Connections:
// Feather GND to switch GND
// Feather 3v3 to switch Headlamp
// Feather pin 11 to switch +

//When switch is ON, digital pin will read LOW

const int SWITCHPIN = 11;     // the number of the toggle switch pin to read
const int LEDPIN = 13;        // the number of the LED pin (using the onboard LED)
int switchState = 0;          // variable for storing the switch state

void setup() {
  // initialize the LED pin as an output
  pinMode(LEDPIN, OUTPUT);
  // initialize the switch pin as an input_pulldown
  pinMode(SWITCHPIN, INPUT_PULLDOWN);
}

void loop() {
  // read the state of the switch value
  switchState = digitalRead(SWITCHPIN);

  // check if the switch is flipped on
  // if it is, the switchState is LOW
  if (switchState == LOW) {
    // turn LED on
    digitalWrite(LEDPIN, HIGH);
  } else { // switchState is HIGH, which is flipped to off
    // turn LED off
    digitalWrite(LEDPIN, LOW);
  }
}
```

To use the switch as a toggle, with the built-in switch LED going from unlit to lit as the switch is flipped from OFF to ON

• connect Feather GND pin to switch GND contact
• connect Feather 3v3 to switch + contact
• connect Feather digital pin, such as pin 11, to switch Headlamp contact
• toggle the switch to send digital pin reading from LOW to HIGH

Here's a simple example sketch to use when testing the switch in LED toggle mode.
//Illuminated toggle switch
//toggle LED on/off mode
//Connections:
// Feather GND to switch GND
// Feather 3v3 to switch +
// Feather pin 11 to switch Headlamp

//When switch is ON, digital pin will read HIGH

const int SWITCHPIN = 11;       // the number of the toggle switch pin to read
const int LEDPIN = 13;          // the number of the LED pin (using the onboard LED)
int switchState = 0;            // variable for storing the switch state

void setup() {
  // initialize the LED pin as an output
  pinMode(LEDPIN, OUTPUT);
  // initialize the switch pin as an input_pulldown
  pinMode(SWITCHPIN, INPUT_PULLDOWN);
}

void loop() {
  // read the state of the switch value
  switchState = digitalRead(SWITCHPIN);

  // check if the switch is flipped on
  // if it is, the switchState is HIGH
  if (switchState == HIGH) {
    // turn LED on
    digitalWrite(LEDPIN, HIGH);
  } else { // switchState is LOW, which is flipped to off
    // turn LED off
    digitalWrite(LEDPIN, LOW);
  }
}

Wire the Switches

Using four 3" lengths of black and red wires, solder the five switches together so they can share a single power and ground connection on the Feather. Solder the black wires between the switches' ground terminals and the red wires between their + terminals.

Solder a 6" length of color matched wire to each switches' headlamp terminal.
ON/OFF Circuit

On a typical project you can power the Feather on and off by connecting its En(able) pin to its GND pin through a latching switch. However, the Crypto Countdown Case is no typical project! You don't want players to be able to power it off once the countdown has begun. Transistor to the rescue.

You can use a simple transistor and two resistors switching circuit to enable and disable the power switch in software. You'll connect the transistor's collector pin to the En(able) pin on the Feather, the transistor's emitter pin to GND and the transistor's base pin to two resistors -- a 10K resistor to the switch (which in turn is connected to
the Feather's VBAT pin) and a 1K resistor which is connected to an output control pin on the Feather, in this case A5 set to OUTPUT mode in software.

When everything is turned off, pressing the on/off button can open and close the Feather's En pin, thus allowing the button to power the Feather on and off. In the Arduino sketch's setup function you can command the A5 pin to go LOW immediately, which in effect disconnects the on/off button from the circuit, thwarting all cheaters who would try to turn it off!

Later in the software, once the case has either been decoded or failed, the software sets the A5 pin back to HIGH, re-enabling the on/off button.

Place the transistor (an EBC pinout, PN2222 transistor in this case), the two resistors, and wires into the holes on the TerminalBlock breakout FeatherWing's prototyping area as shown, and solder them together on the back side. You can follow the breadboarded diagram as a guide.
To enable an easy connection between the power button and the switching circuit, solder in a 2-pin JST connector: one wire to the BAT pin solder point and the other wire to the 10K resistor.

You'll solder the other side of the JST cable to the COMMON and N.O. terminals of the latching button.
Display

Solder in the stacking headers on the Feather M0 board instead of the normal headers. This way you'll be able to plug the Feather into the Terminal Block breakout FeatherWing and the display FeatherWing can go on top of the Feather. It's a delightful Feather sandwich! (Ptoo.)

Assemble the 14-segment display shield as instructed here, then stack it onto the Feather, and the stack the Feather onto the TerminalBlock breakout FeatherWing.
Power Button Lighting

The latching power button has a built-in LED to give it its green glow. You'll run that LED from the Feather output pin 13. Solder an 8" length of red wire to the button's + terminal and black wire to its GND terminal.

NeoPixel Indicator Lamp

The 5mm NeoPixel has four legs, Data In, +, GND, and Data Out. We don't need the data out capability, so you can trim that leg. Then, prepare three female jumper wires
to a 3-row wire housing by pushing their crimp connectors into the housing. Trim and strip the wires on the other side, these will be pulled into the screw terminals.
Note the position of the long leg of the NeoPixel -- this is GND. Make a black mark on one wire at both ends and plug the GND leg into this space on the connector to keep track of the wiring order.

**BEEEEEP**

You can now connect the piezo buzzer to the screw terminals. Strip a bit of insulation from the ends of each wire on the buzzer and then screw the red wire into A0 and the black wire into a GND terminal.
Wiring

Make the remaining wiring connections as seen in the diagram. Since the on/off button will need to be threaded through the case panel later, that one will need to be disconnected and then reconnected, but for now it's good to connect everything for testing before everything is mounted.

Connect the following from component to screw terminals:

Switches

- Switch bundle's + to 3v3
- Switch bundle's black wire to GND
- White switch's white wire (headlamp) to 12
- Red switch's orange wire to 11
- Yellow switch's yellow wire to 10
- Green switch's green wire to 6
- Blue switch's blue wire to 5

Power Button

- Power button red wire to pin 13
- Power button black wire to GND
- Power button JST to switch transistor circuit JST

NeoPixel

- NeoPixel GND to GND
- NeoPixel + to 3v3
- NeoPixel Data In to A2
Upload the Code

With the system assembled, you can now upload the Arduino code to the Feather M0 for testing and configuration.

If you are new to the Feather M0, [start here](#) to set up the board for use with the Arduino IDE.

When you're comfortable uploading code, such as a basic Blink sketch, to your Feather, you can now install the libraries used in the Crypto Countdown Case.

In the Arduino IDE, click on Sketch > Include Library> Manage Libraries... to open up the Library Manager. From here, search for these three libraries and then click the Install button for each:

- Adafruit GFX
- Adafruit BusIO
- Adafruit LED Backpack
- Adafruit NeoPixel

Once those are installed, close the Library Manger and download the following sketch, uncompress it, and place it in your Arduino sketches directory. Open the sketch in the Arduino IDE and upload it to your Feather M0.

![CryptoCountdownCase.zip](#)

Now to test it out. Disconnect the Feather from the computer USB, then plug in the battery to the Feather. Make sure the TerminalBlock FeatherWing's on-board switch is in the ON position, then press the green power button. It will light up and send the display and buzzer into a boot sequence, blink the time at "03.00" and then start the countdown. Flip the toggles in the proper order -- red, white, blue, green yellow -- and the display will freeze, scroll "Disarming", and give the secret coded message.

Reset the board, and try it again, but enter an incorrect password this time. The display will quickly run down to 00.00 and display the "X.X.X.X." and "Boom" messages, while beeping.

There are a number of places in the code to customize things. You can:

- Change the allowReset variable from 1 to 0 in order to prevent players from turning off the box
- Use a different toggle switch order by changing the PASSWORD[] array
• Adjust the countdown duration with the gTimer variable (be sure to change the displayScroll message and blink text in the setup loop to match)
• Give the players a customized message after successfully entering the password in the section following the displayScroll(" Disarmed ");

//Crypto Countdown Case
//by John Park and Usman Muzaffar
//for Adafruit Industries
//MIT License, include above text in redistribution

#include <Wire.h>
#include <Adafruit_GFX.h>
#include "Adafruit_LEDBackpack.h"
#include <Adafruit_NeoPixel.h>
#define NEOPIN            A2  //NeoPixel attached to pin A2 of the Feather
#define BUZZPIN           A0  //Piezo buzzer attached to Feather pin A0 //dev hack, set to A1 for quiet
const int LEDPIN = 13;

int allowReset = 1;  //set to 0 to disable user reset with the power button once started  
                      //if set to 0, wait until end of game, turn off all toggles, then turn off power
Adafruit_NeoPixel pixel = Adafruit_NeoPixel(1, NEOPIN, NEO_RGB + NEO_KHZ800);
Adafruit_AlphaNum4 alpha4 = Adafruit_AlphaNum4();

// Password combo variables
const int PASSWORD[5]={2,1,5,4,3}; //stores the password set
int inputCode[5];  //stores the input code set
int entered = 0; //indicates a full code set has been entered
int over = 0;  //state of game for escaping while loop
int blinker = 0;  //for decimal point blinking in updateDisplay function

// Switch variables
const int WHITESWITCHPIN = 12;  // using illuminated toggle switch, Pin to Headlamp, 3v3 to +, GND to GND, use INPUT_PULLDOWN
const int REDSWITCHPIN = 11;
const int YELLOWSWITCHPIN = 10;
const int GREENSWITCHPIN = 6;
const int BLUESWITCHPIN = 5;

int whiteSwitchState = 0;  //variable to read switch state, off/0 or on/1
int redSwitchState = 0;
int yellowSwitchState = 0;
int greenSwitchState = 0;
int blueSwitchState = 0;

int whiteSwitchPriorState = 0;  //variable to store switch last state
int redSwitchPriorState = 0;
int yellowSwitchPriorState = 0;
int greenSwitchPriorState = 0;
int blueSwitchPriorState = 0;

int fail = 0;  // if fail != 0, the combination failed
int success = 0;  // counter for successful entry of combination
int count = 0;  // number of switches flipped

#include <Adafruit_GFX.h>
const unsigned long FAST = 30; // 100 this is the rate for fast countdown when combo
fails, smaller is faster
unsigned int gDuration = NORMAL;

bool updateAndCheckTimer() {
    static unsigned long now, lastUpdated = 0;
    now = millis();
    if (gTimer && now - lastUpdated > gDuration) {
        gTimer--;
        lastUpdated = now;
        return true;
    }
    return false;
}

void updateDisplay() {
    // Given gTimer, update the alpha4 display
    char minutesStr[10], secondsStr[10];
    int minutes = gTimer / 60;
    int seconds = gTimer - (minutes * 60);
    sprintf(minutesStr, "%02d", minutes);
    sprintf(secondsStr, "%02d", seconds);
    Serial.print(minutesStr[0]);
    Serial.print(minutesStr[1]);
    Serial.print('.');
    Serial.print(secondsStr[0]);
    Serial.print(secondsStr[1]);
    Serial.println();
    alpha4.writeDigitAscii(0, minutesStr[0], 0);
    // blinking the dot on second character
    if (blinker == 0) {
        alpha4.writeDigitAscii(1, minutesStr[1], 0);
        blinker = 1;
        noTone(BUZZPIN);
    } else if (blinker == 1) {
        alpha4.writeDigitAscii(1, minutesStr[1], 1);
        blinker = 0;
        tone(BUZZPIN, 350);
    }
    alpha4.writeDigitAscii(2, secondsStr[0], 0);
    alpha4.writeDigitAscii(3, secondsStr[1], 0);
    alpha4.writeDisplay();
    if (gDuration == FAST) {
        tone(BUZZPIN, 700);
    }
    if ((minutes == 0) && (seconds == 0)) { // time ran out
        fail = 1;
    }
}

void displayScroll(const char*s) {
    const int NUMCHARS = 4;
    int scrollRate = 400; // delay time in milis for character fill in rate
    int len = strlen(s);
    for (int i = 0; i < len - NUMCHARS; i++) {
        for (int j = 0; j < NUMCHARS; j++) {
            alpha4.writeDigitAscii(j, s[i + j]);
        }
    }
    alpha4.writeDisplay();
    tone(BUZZPIN, 100);
    delay(scrollRate);
    // noTone(BUZZPIN);
void setup() {
  if (allowReset==0){
    pinMode(A5, OUTPUT);//disables the on/off button
    digitalWrite(A5, LOW);//disables the on/off button
  }
  //gTimer = 180;           //how much time the player has to enter the code
  gTimer = 180; // countdown time in seconds
  pixel.begin();
  pixel.setBrightness(5);
  pixel.setPixelColor(0, 200, 0, 0);//starts red
  pixel.show();
  pinMode(LEDPIN, OUTPUT);
  pinMode(WHITESWITCHPIN, INPUT_PULLDOWN); // this is for M0 SAMD boards, on
  other boards use 10K resistor and INPUT mode
  pinMode(REDSWITCHPIN, INPUT_PULLDOWN);
  pinMode(YELLOWSWITCHPIN, INPUT_PULLDOWN);
  pinMode(GREENSWITCHPIN, INPUT_PULLDOWN);
  pinMode(BLUESWITCHPIN, INPUT_PULLDOWN);
  Serial.begin(9600); // for debugging only
  alpha4.begin(0x70);  // pass in the address
  // fill the characters with a flicker
  int b = 0;
  for(b=1; b&lt;=16; b=b+5){
    alpha4.clear();
    alpha4.writeDisplay();
    delay(50);
    alpha4.setBrightness(b);
    alpha4.writeDigitRaw(0, 0xFFFF);
    alpha4.writeDigitRaw(1, 0xFFFF);
    alpha4.writeDigitRaw(2, 0xFFFF);
    alpha4.writeDigitRaw(3, 0xFFFF);
    alpha4.writeDisplay();
    delay(200);
  }
  delay(1500);
  int fillRate = 200; // delay time in millis for character fill-in rate
  // scroll "---Arming---" on display
  displayScroll("---Arming----- **** 0300 ");
  delay(1000);
  alpha4.clear(); // wipe the display
  alpha4.writeDisplay();
  delay(500);
  // blink the time 3 times while beeping
  for (int i=0; i&lt;3; i++) {
    alpha4.clear();
    alpha4.writeDisplay();
    delay(300);
    alpha4.writeDigitAscii(0, '0', 0);
    alpha4.writeDigitAscii(1, '3', 1);
    alpha4.writeDigitAscii(2, '0', 0);
    alpha4.writeDigitAscii(3, '0', 0);
    alpha4.writeDisplay();
    tone(BUZZPIN, 240);
    delay(625);
    noTone(BUZZPIN);
  }
  // hold time before countdown begins
  alpha4.clear();
  alpha4.writeDisplay();
  delay(300);
alpha4.writeDigitAscii(0, '0', 0);
alpha4.writeDigitAscii(1, '3', 1);
alpha4.writeDigitAscii(2, '0', 0);
alpha4.writeDigitAscii(3, '0', 0);
alpha4.writeDisplay();
tone(BUZZPIN, 480);
delay(2000);
noTone(BUZZPIN);

pixel.setBrightness(20);
pixel.setPixelColor(0, 255, 153, 0);//goes to yellow
pixel.show();
}
void loop() {
  if (updateAndCheckTimer()) {
    updateDisplay();
  }

  //check the switches to see if they're toggled
  whiteSwitchState = digitalRead(WHITESWITCHPIN);
  redSwitchState = digitalRead(REDSWITCHPIN);
  yellowSwitchState = digitalRead(YELLOWSWITCHPIN);
  greenSwitchState = digitalRead(GREENSWITCHPIN);
  blueSwitchState = digitalRead(BLUESWITCHPIN);

  if ((whiteSwitchState==HIGH)&&(whiteSwitchPriorState==0)){
    inputCode[(count)]=1;
    count=count+1;
    whiteSwitchPriorState=1;
    Serial.print("white flipped, count is ");
    Serial.println(count);
    pixel.setPixelColor(0, 255, 255, 255);//white
    pixel.show();
    tone(BUZZPIN, 260);
    delay(150);
    noTone(BUZZPIN);
  }

  if ((redSwitchState==HIGH)&&(redSwitchPriorState==0)){
    inputCode[(count)]=2;
    count=count+1;
    redSwitchPriorState=1;
    Serial.print("red flipped, count is ");
    Serial.println(count);
    pixel.setPixelColor(0, 255, 0, 0);//white
    pixel.show();
    tone(BUZZPIN, 260);
    delay(150);
    noTone(BUZZPIN);
  }

  if ((yellowSwitchState==HIGH)&&(yellowSwitchPriorState==0)){
    inputCode[(count)]=3;
    count=count+1;
    yellowSwitchPriorState=1;
    Serial.print("yellow flipped, count is ");
    Serial.println(count);
    Serial.println(success);
    pixel.setPixelColor(0, 255, 255, 0);//white
    pixel.show();
    tone(BUZZPIN, 260);
    delay(150);
    noTone(BUZZPIN);
  }

  if ((greenSwitchState==HIGH)&&(greenSwitchPriorState==0)){
    inputCode[(count)]=4;
    count=count+1;
    greenSwitchPriorState=1;
Serial.print("green flipped, count is ");
Serial.println(count);
pixel.setPixelColor(0, 0, 255, 0);//white
pixel.show();
tone(BUZZPIN, 260);
delay(150);
noTone(BUZZPIN);
}

if ((blueSwitchState==HIGH)&&(blueSwitchPriorState==0)) {
  inputCode[(count)]=5;
  count=count+1;
  blueSwitchPriorState=1;
  Serial.print("blue flipped, count is ");
  Serial.println(count);
pixel.setPixelColor(0, 0, 255, 0);//white
pixel.show();
tone(BUZZPIN, 260);
delay(150);
noTone(BUZZPIN);
}

if (count==5) {
  entered=1;
}

while ((entered!=0)&&(over==0)) {
  if (inputCode[0]==PASSWORD[0]&&
      inputCode[1]==PASSWORD[1]&&
    success=1;
    //flash the final number
    for(int j=0;j<3;j++) {
      Serial.println("You Did It");
      Serial.println(gTimer);
tone(BUZZPIN, 349);
updateDisplay();
delay(700);
noTone(BUZZPIN);
alpha4.clear();
alpha4.writeDisplay();
delay(700);
    }
    over=1;
  } else {
    fail=1;
    over=1;
    Serial.println("failed, fast timer now");
gDuration = FAST;
  }
}

while (success==1)//after the correct code has been entered, do this
  //updateDisplay();
delay(2000);
pixel.setBrightness(30);
pixel.setPixelColor(0, 0, 225, 0);//goes to green
pixel.show();
alpha4.clear();
alpha4.writeDisplay();
int fillRate = 200; //delay time in milis for character fill in rate
displayScroll(" Disarmed ");
noTone(BUZZPIN);
alpha4.clear(); //wipe the display
alpha4.writeDisplay();
//scroll L to R filled symbols while beeping with each landing
// change each character for a different message, use writeDigitAscii for regular alphanumerics
//scroll "ZHE"
alpha4.writeDigitRaw(3, 0x0);
alpha4.writeDigitRaw(0, 0x3F00);
alpha4.writeDisplay();
delay(500);
alpha4.writeDigitRaw(0, 0x0);
alpha4.writeDigitRaw(1, 0x3F00);
alpha4.writeDisplay();
delay(500);
alpha4.writeDigitRaw(1, 0x0);
alpha4.writeDigitRaw(2, 0x3F00);
alpha4.writeDisplay();
delay(500);
alpha4.writeDigitRaw(2, 0x0);
alpha4.writeDigitRaw(3, 0x7F00); //add decimal point
alpha4.writeDisplay();
tone(BUZZPIN, 400);
delay(500);
noTone(BUZZPIN);
delay(500);
//scroll "B"
alpha4.writeDigitRaw(0, 0xFD);
alpha4.writeDisplay();
delay(500);
alpha4.writeDigitRaw(0, 0x0);
alpha4.writeDigitRaw(1, 0xFD);
alpha4.writeDisplay();
delay(500);
alpha4.writeDigitRaw(1, 0x0);
alpha4.writeDigitRaw(2, 0x40FD); //add decimal point
alpha4.writeDisplay();
tone(BUZZPIN, 400);
delay(500);
noTone(BUZZPIN);
delay(500);
//scroll "G"
alpha4.writeDigitRaw(0, 0x31);
alpha4.writeDisplay();
delay(500);
alpha4.writeDigitRaw(0, 0x0);
alpha4.writeDigitRaw(1, 0x4031);
alpha4.writeDisplay();
tone(BUZZPIN, 400);
delay(500);
noTone(BUZZPIN);
delay(500);
//"K"
alpha4.writeDigitAscii(0, 'K', 1);
alpha4.writeDisplay();
tone(BUZZPIN, 400);
delay(1000);
noTone(BUZZPIN);
delay(4000);

//check the switches re-enable power switch to turn it off
whiteSwitchState = digitalRead(WHITESWITCHPIN);
redSwitchState = digitalRead(REDSWITCHPIN);
yellowSwitchState = digitalRead(YELLOWSWITCHPIN);
greenSwitchState = digitalRead(GREENSWITCHPIN);
blueSwitchState = digitalRead(BLUESWITCHPIN);

if ((whiteSwitchState==LOW) && (redSwitchState==LOW) &&
(yellowSwitchState==LOW) && (greenSwitchState==LOW) &&
(blueSwitchState==LOW)){
digitalWrite(A5, HIGH);//re-enable the on/off button
}
}

while (!gTimer && fail==1){ //after incorrect code is entered, do this
  pixel.setBrightness(50);
  pixel.setPixelColor(0, 255, 0, 0);//set to red
  pixel.show();
  Serial.println("failed");
  alpha4.writeDigitAscii(0, 'X', 1);
  alpha4.writeDigitAscii(1, 'X', 1);
  alpha4.writeDigitAscii(2, 'X', 1);
  alpha4.writeDigitAscii(3, 'X', 1);
  alpha4.writeDisplay();
  tone(BUZZPIN, 900);
  delay(500);
  over=1;

  pixel.setBrightness(100);
  pixel.setPixelColor(0, 255, 0, 0);
  pixel.show();
  alpha4.writeDigitAscii(0, 'B');
  alpha4.writeDigitAscii(1, 'o');
  alpha4.writeDigitAscii(2, 'o');
  alpha4.writeDigitAscii(3, 'm');
  alpha4.writeDisplay();
  delay(200);
  pixel.setPixelColor(0, 0, 0, 0);
  pixel.show();
  alpha4.clear();
  alpha4.writeDisplay();
  tone(BUZZPIN, 800);
  delay(500);
  noTone(BUZZPIN);

  //check the switches re-enable power switch to turn it off
  whiteSwitchState = digitalRead(WHITESWITCHPIN);
  redSwitchState =    digitalRead(REDSWITCHPIN);
  yellowSwitchState = digitalRead(YELLOWSWITCHPIN);
  greenSwitchState =  digitalRead(GREENSWITCHPIN);
  blueSwitchState =   digitalRead(BLUESWITCHPIN);

  if (((whiteSwitchState==LOW)&amp;&amp;(redSwitchState==LOW)&amp;&amp;&amp;
         (yellowSwitchState==LOW)&amp;&amp;&amp;(greenSwitchState==LOW)&amp;&amp;&amp;
         (blueSwitchState==LOW)){
    digitalWrite(A5, HIGH);//re-enable the on/off button
  }
}

Next, let's put it all inside a suitably themed, Cold War-era style enclosure.
Make the Case

Distressing

Your Crypto Countdown circuit deserves to live in a rugged-yet-age-worn case to give it that action thriller feel. You can find a vintage case that's ready to go, or start with a brand new case and distress it with watered-down paint.

With the case closed, apply a heavily watered down black acrylic craft paint wash with a brush and let it dry overnight.
You can wipe off paint in some areas, and leave it nice and dark in the recesses and crevices.

To go with the Soviet KGB theme, you can spray paint a red star onto the case using a stencil. I’ve included a star template in the files linked to here, which you can print and cut out to trace onto mylar or cardboard stencil material.

Start off with some gray primer, then once that has dried, spray on your logo.
If your case came with pick-and-place foam, you can remove it entirely, or cut it out to fit the parts.

Recharge the lipoly battery by simply plugging a USB cable from the case port to a charger wall wart or computer.
Mounting Panel

The switches, display, buzzer, indicator LED, USB port, and power button all need to be mounted to a panel. You can do this in a variety of ways -- I started out with hand cut cardboard for prototyping, then laser cut cardboard, baltic birch plywood and finally acrylic for the final prop.

There are a variety of ways to cut the panel, from hobby knives, to hand-held coping saws and drills, to bandsaw, CNC machines, and laser cutters. For highly professional pieces, you can have custom panels made by a service bureau or online panel maker.

Download the files linked here and print them out as a template for cutting, or to drive a laser cutter or CNC. Or, use it as a guide to model a file for 3D printing.

box4SVG.zip
In order to add the graphics I ran a second etching pass on the laser cutter to lightly score the acrylic, then painted them with white acrylic craft paint and immediately wiped it off. The paint comes off of the smooth surface easily but stays in the rough etched portion of the graphics.

I also ran a third, low speed and high power etching pass on the section that holds the missile switch safety covers in place. This prevents them from twisting around at all. Alternately you can add a bit of glue to hold them in place, or turn the nut hard enough to bite into a softer material such as wood or cardboard.

For added support to keep the panel from flexing I added a second, inner panel with cutouts to clear all parts and screwed it to the top panel using the 2.5mm nylon screws and nuts.
Mounting the Parts

Now you can mount the parts in your panel. Remove the top nut and safety cover from each switch, turn the lower nut up to the midway point of the shaft and then push the each toggle switch through its respective panel hole, lining up the thread groove with the panel hole tab.

Then, place the safety cover over each switch shaft, and tighten the top nut into place.
Standoffs

You'll mount the FeatherWing-Feather-FeatherWing sandwich to the panel using nylon standoffs. Make an appropriately sized standoff by screwing together as many as needed to mount the display flush with the panel.

Then, fasten them to the panel with screws from the top, leaving the threaded standoff ends to affix the FeatherWing.
Test the FeatherWing mounting position but don't screw the nuts into place yet, you'll still need it free to connect the power button wiring.
Power Button

Unthread the illuminated on/off power button's locking collar and then feed the wires and JST cable through its hole on the top-side of the panel.

Then, screw the collar back on, securing it in place.
Re-attach the power button's wires to the screw terminals (this is easier to do before the FeatherWing is mounted) and re-connect the JST connectors.
LED Indicator

Place the NeoPixel LED into its holder, then push the holder into the panel from the top side. Then connect the 3-pin connector to the LED's legs, being sure to mind the color order. Remember, the long leg is the GND terminal.

You can gently bend the legs to a 90 degree angle if you like for better clearance with the case foam.
Piezo Mounting

Mount the piezo buzzer to the case using two 2.5mm nylon screws and nuts.
USB Connector

For re-programming and battery charging (the Feather has built-in charging circuitry), you'll want to be able to still plug the Feather into USB after the case is assembled. The panel mount USB connector allows us to do just that!

Fasten the USB panel mount using two 2.5mm nylon screws and nuts, then plug the male end into the Feather M0.
Final Steps

Screw the FeatherWing sandwich onto the standoffs with the four nylon nuts. Then, connect the battery, and place the panel into the case. This will be secured by a nice, tight friction-fit -- just be sure escape room players know they aren't meant to disassemble props to solve them! You can also choose to more permanently seal the panel with glue or fasteners.
Deploy the Case

You can charge the battery by simply plugging in USB power over the the case's USB jack, thanks to the built in charging circuit of the Feather.

Usage

The moment has arrived, time to put the Crypto Countdown Case into action! Press the tempting green button to turn it on, watch the ominous boot sequence, begin to feel the sweat drip down your forehead as the timer counts relentlessly down, and begin to enter your disarm sequence of toggle switches!
Whew! You've saved the free world!! Write down the secret message, turn off the toggles, power down the case, and move on to your next assignment.