



video

https://youtu.be/PbaC4wB-rKM?si=COJ_ag-AEVTXa98b

<https://nyos.dev/prescript>

PARTS LIST

1. MICROCONTROLLER

ESP32

2. DISPLAY

ILI9341

3. POWER SUPPLY

18650 accumulator (1-2)

TP4056 (USB-C ,microUSB)

DC-DC Step Down 5V converter

4. CONNECTIONS

Dupont (male-female / male-male)

breadboard

PCB

5. THINGS

1k Ω resistor (for DFPlayer RX)

1000 μ F capacitor (power supply stabilization)

Buttons (3–5 pcs) for control

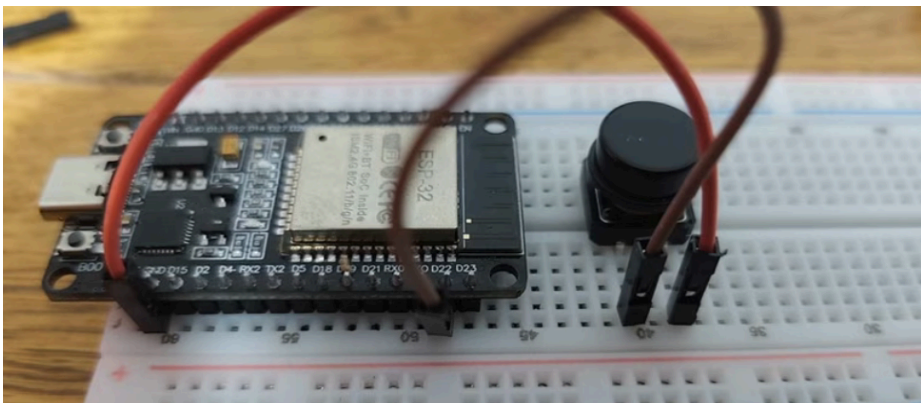
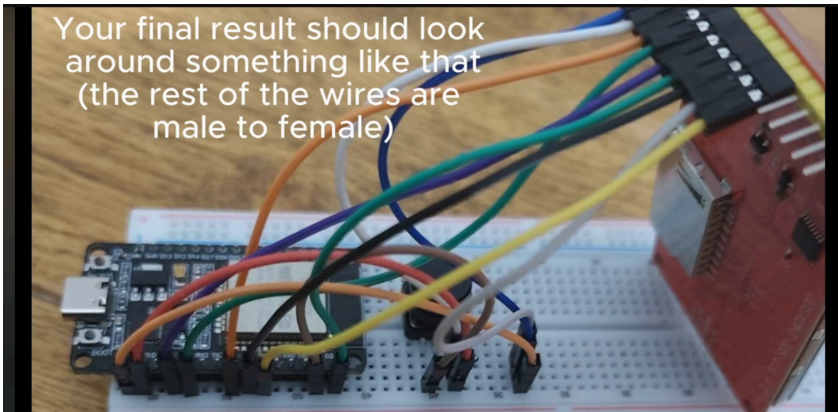
Structure (I have no fucking clue how to do this)

```
Screen          ESP32

VCC ---> 3V3
GND ---> GND
CS  ---> D5
RST ---> D4
DC  ---> D2
MOSI---> D23
SCK ---> D18
LED ---> 3V3
MISO---> D19

Button          ESP32

pin1 ---> GND
pin2 ---> D22
```



Code

////////////////////////////////////

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <HardwareSerial.h>

// first display related pins (on the esp32)
#define TFT_CS 5
#define TFT_DC 2
#define TFT_RST 4

Adafruit_ILI9341 tft(TFT_CS, TFT_DC, TFT_RST);

// pin related to the button (on the esp32)
#define BUTTON_PIN 22 //

// timing variables for the letters that you can change to fit your needs (in milliseconds)
#define SCRAMBLE_DURATION 800 // the randomizing/scrambling will last 800
milliseconds before freezing each letter one by one
#define FREEZE_DELAY 100 // the freeze delay between each letter after the
scrambling ends
#define RANDOM_INTERVAL 15 // interval before a special character/letter/number gets
updated

char letters[8]; // Makes it so 8 characters show up
bool stopped[8]; // Makes it track which character stops scrambling

unsigned long startTime = 0;
unsigned long lastFreezeTime = 0;
unsigned long lastRandomTime = 0;

bool animating = false;
bool freezing = false;
int freezeIndex = 0;
int pressCount = 0;
```

```

int specialOutcome = 0; // 0=normal, 1=im6rburG, 2=JACKPOT, 3=CLEAR, you get the gist
int xStart; // where the text will show up on screen on the x axis, calculated later
int yStart = 100; // same but for the y axis

uint16_t HERMES_COLOR;

// basically makes it so lowercase, uppercase and digits are used

char randomChar() {
    int r = random(0, 62);
    if (r < 26) return 'A' + r;
    else if (r < 52) return 'a' + (r - 26);
    else return '0' + (r - 52);
}

// main function that sets the base layout for the animation

void resetAnimation() {
    for (int i = 0; i < 8; i++) {
        stopped[i] = false;
        letters[i] = randomChar();
    }

    freezeIndex = 0;
    freezing = false;
    animating = true;

    // special words that appear on random chances
    int chance = random(1, 101); // random number from 1 to 100 , since 100 is the rarest

    if (chance == 1) { // 1/100 chance for JACKPOT!
        specialOutcome = 2;
    }
    else if (chance <= 7) { // ~1/15 chance (roughly 7/100) for im6rburG
        specialOutcome = 1;
    }
    else if (pressCount == 3) { // _CLEAR._ on 3rd press
        specialOutcome = 3;
    }
    else {
        specialOutcome = 0; // normal random chars freeze
    }

    startTime = millis();
    lastRandomTime = millis();

```

```
tft.fillRect(xStart, yStart - 5, 320, 60, ILI9341_BLACK); // clear text with the stored variable
(black background)
tft.setCursor(xStart, yStart);
tft.setTextColor(HERMES_COLOR, ILI9341_BLACK); // sets the color of the characters to
the blue letter that's defined later on in the script
tft.setTextSize(6);
```

```
for (int i = 0; i < 8; i++) tft.print(letters[i]); // shows the initial characters before randomizing
}
```

```
// randomizing characters function
```

```
void updateLettersFast() {
  tft.setTextColor(HERMES_COLOR, ILI9341_BLACK); // randomizes to the blue color while
  keeping the black background
  tft.setTextSize(6); // text size, you can change it however you want
```

```
for (int i = 0; i < 8; i++) {
  if (!stopped[i]) {
    char oldChar = letters[i];
    letters[i] = randomChar();

    if (letters[i] != oldChar) {
      int xChar = xStart + (i * 36);
      tft.setCursor(xChar, yStart);
      tft.print(letters[i]);
    }
  }
}
}
```

```
//sets up the base led screen stuff
```

```
void setup() {
```

```
  pinMode(BUTTON_PIN, INPUT_PULLUP);
```

```
  SPI.begin(18, 19, 23, 5); // initializes the communication with the pins listed here (pins of
the esp32 still)
```

```
  tft.begin();
  tft.setRotation(1);
  tft.fillRect(ILI9341_BLACK);
```

```
  HERMES_COLOR = tft.color565(0x75, 0xA0, 0xE3); // this is where the characters color is
defined, so feel free to change it to your convenience
```

```
int textWidth = 8 * 36; // calculates 8 characters times 36 pixels
xStart = (320 - textWidth) / 2; // centers the characters according to the screen's width (320
pixels here)
```

```
randomSeed(esp_random());
```

```
delay(200);
}
```

```
// loop function that triggers with the button
```

```
void loop() {
```

```
if (digitalRead(BUTTON_PIN) == LOW && !animating) {
  delay(150); //
  pressCount++;
  if (pressCount > 3) pressCount = 1;
```

```
  resetAnimation();
  updateLettersFast();
}
```

```
if (!animating) return;
```

```
unsigned long currentTime = millis();
```

```
if (currentTime - lastRandomTime >= RANDOM_INTERVAL) {
  updateLettersFast();
  lastRandomTime = currentTime;
}
```

```
if (!freezing && currentTime - startTime >= SCRAMBLE_DURATION) {
  freezing = true;
  lastFreezeTime = currentTime;
}
```

```
if (freezing && freezeIndex < 8) {
  if (currentTime - lastFreezeTime >= FREEZE_DELAY) {
```

```
    // chooses the previously set up special outcome
    if (specialOutcome == 2) { // JACKPOT!
      char jackpotText[8] = {'J', 'A', 'C', 'K', 'P', 'O', 'T', '!'};
      letters[freezeIndex] = jackpotText[freezeIndex];
      int xChar = xStart + (freezeIndex * 36);
      tft.setCursor(xChar, yStart);
      tft.print(letters[freezeIndex]);
      tft.setTextColor(HERMES_COLOR, ILI9341_BLACK); // resets color
```

```

}
else if (specialOutcome == 1) { // im6rburG
    char burgerText[8] = {'i', 'm', '6', 'r', 'b', 'u', 'r', 'G'};
    letters[freezeIndex] = burgerText[freezeIndex];
    int xChar = xStart + (freezeIndex * 36);
    tft.setCursor(xChar, yStart);
    tft.print(letters[freezeIndex]);
}
else if (specialOutcome == 3) { // _CLEAR._
    char finalText[8] = {'_', 'C', 'L', 'E', 'A', 'R', '!', '_'};
    letters[freezeIndex] = finalText[freezeIndex];
    int xChar = xStart + (freezeIndex * 36);
    tft.setCursor(xChar, yStart);
    tft.print(letters[freezeIndex]);
}
// else, normal random freeze (no special text like JACKPOT! or the others)

stopped[freezeIndex] = true;
freezeIndex++;
lastFreezeTime = currentTime;

if (freezeIndex >= 8) animating = false; // stops the animation
}
}
}

```