#include <Arduino.h>

#include "Setup1\_st7789\_130x320.h"

#include <WiFi.h>

#include "offgrid\_intro.h"

// Paramètres WiFi

const char\* ssid = "RadissonBlu-Guest";

// couleur de fond pour TFT espi

uint16\_t color = tft.color565(255, 220, 58);

uint16\_t colorfond = tft.color565(19, 19, 19);

uint16\_t colorgrey = tft.color565(194, 194, 194);

const char\* keys[] = {

  "1.,?!@", "2ABC", "3DEF", "4GHI", "5JKL", "6MNO", "7PQRS", "8TUV", "9WXYZ", "\*SHIFT", "0", "#"

};

const int gpioKeys[] = {

  7, 17, 18, 8, 46, 9, 10, 11, 12, 13, 21, 47

};

bool isShift = false;

String edittext(String zetext) {

  int lastKeyIndex = -1;

  int countKeyPress = 0;

  unsigned long lastKeyPressTime = 0;

  const unsigned long timeout = 1000; // 1 second to switch character or confirm

  while (true) {

    for (int i = 0; i < 12; i++) {

      if (digitalRead(gpioKeys[i]) == HIGH) { // Assuming pull-up configuration, button press is LOW

        if (i == 9) { // SHIFT key

          isShift = !isShift;

        } else if (i == 10) { // BACKSPACE key

          if (zetext.length() > 0) {

            zetext.remove(zetext.length() - 1);

          }

        } else if (i == 11) { // ENTER key

          return zetext;

        } else {

          if (lastKeyIndex == i && (millis() - lastKeyPressTime) < timeout) {

            countKeyPress++;

          } else {

            countKeyPress = 0;

          }

          lastKeyIndex = i;

          lastKeyPressTime = millis();

          int charIndex = countKeyPress % strlen(keys[i]);

          char character = keys[i][charIndex];

          if (isShift && character >= 'a' && character <= 'z') {

            character -= 32; // Convert to uppercase

          }

          zetext += character;

          if (countKeyPress > 0 && zetext.length() > 1) {

            zetext.remove(zetext.length() - 2, 1); // Remove the previous character

          }

        }

        // Display the updated zetext

        tft.fillScreen(TFT\_BLACK);

        tft.pushImage(0, 30, 320, 172, offgrid\_intro);

        tft.setCursor(10, 100);

        tft.setTextColor(color);

        tft.setTextSize(2);

        tft.print(zetext);

        // Delay to manage button debounce

        delay(200);

      }

    }

  }

}

void setup() {

  pinMode(TFT\_PWR\_PIN, OUTPUT);

  pinMode(TFT\_BL\_PIN, OUTPUT);

  digitalWrite(TFT\_PWR\_PIN, HIGH);

  digitalWrite(TFT\_BL\_PIN, HIGH);

  tft.begin();

  tft.setRotation(3);

  tft.setSwapBytes(true);

  tft.fillScreen(TFT\_BLACK);

  tft.pushImage(0, 30, 320, 172, offgrid\_intro);

  for (int i = 0; i < 12; i++) {

    pinMode(gpioKeys[i], INPUT\_PULLUP);

  }

  WiFi.begin(ssid);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(1000);

    Serial.println("Connexion au WiFi...");

  }

}

void loop() {

  // For demonstration purposes:

  String result = edittext("");

  tft.setCursor(10, 130);

  tft.setTextColor(colorgrey);

  tft.setTextSize(2);

  tft.print(result);

  delay(5000);  // Display the result for 5 seconds before clearing

}