

#include "SPI.h"

#include "TFT\_22\_ILI9225.h"

#define TFT\_RST 2

#define TFT\_RS  4

#define TFT\_CS  15  // SS

#define TFT\_SDI 13  // MOSI

#define TFT\_CLK 14  // SCK

#define TFT\_LED 0   // 0 if wired to +5V directly

// Use software SPI (slower)

TFT\_22\_ILI9225 tft = TFT\_22\_ILI9225(TFT\_RST, TFT\_RS, TFT\_CS, TFT\_SDI, TFT\_CLK, TFT\_LED);

const unsigned char gImage\_ic[3872] PROGMEM= { /\* 0X00,0X01,0XB0,0X00,0XB0,0X00, \*/ };

// Setup

void setup() {

  tft.begin();

  Serial.begin(9600);

}

// Loop

void loop() {

  unsigned long startTime = millis(); // declare and initialize startTime

  tft.drawBitmap(0, 0, gImage\_ic, 176, 176, COLOR\_CYAN);

  tft.setBackgroundColor(COLOR\_BLACK);

  tft.setFont(Terminal6x8);

  tft.drawText(0, 178, "Printing some random text.", COLOR\_RED);

  tft.drawText(0, 188, "Another line of nonsense", COLOR\_BLUE);

  unsigned long duration = millis()-startTime; // declare and initialize duration

  tft.drawText(0, 198, String(duration), COLOR\_WHITE);

  while(true);

}

#include <ESP8266WiFi.h>

#include "SPI.h"

#include "TFT\_22\_ILI9225.h"

#include <ArduinoJson.h>

#include <WiFiClientSecure.h>

#include <ArduinoJson.h>

#define TFT\_RST 2

#define TFT\_RS  4

#define TFT\_CS  15  // SS

#define TFT\_SDI 13  // MOSI

#define TFT\_CLK 14  // SCK

#define TFT\_LED 0   // 0 if wired to +5V directly

// Thay đổi các giá trị này bằng thông tin WiFi của bạn

const char\* ssid = "P811";

const char\* password = "anhem811";

// Thay đổi API\_KEY bằng API key của bạn từ OpenWeatherMap

const char\* host = "api.openweathermap.org";

const char\* API\_KEY = "2f60058a0618d731ab76b8b89adb2608";

// Thay đổi tọa độ của bạn bằng tọa độ thực tế của bạn

const float latitude = 37.7749;

const float longitude = -122.4194;

// Khai báo biến cho kết nối WiFi và màn hình TFT

WiFiClientSecure client;

// Use software SPI (slower)

TFT\_22\_ILI9225 tft = TFT\_22\_ILI9225(TFT\_RST, TFT\_RS, TFT\_CS, TFT\_SDI, TFT\_CLK, TFT\_LED);

// Setup

void setup() {

  tft.begin();

  Serial.begin(115200);

  // Kết nối với WiFi

  WiFi.begin(ssid, password);

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

    delay(1000);

    Serial.println("Kết nối đến WiFi...");

  }

  Serial.println("Kết nối thành công!");

  // Kết nối đến API dự báo thời tiết

  client.setInsecure();

  if (!client.connect(host, 443)) {

    Serial.println("Lỗi kết nối đến API dự báo thời tiết!");

  }

}

// Loop

void loop() {

  // Gửi yêu cầu GET để lấy thông tin dự báo thời tiết từ OpenWeatherMap API

  String url = "/data/2.5/weather?lat=" + String(latitude) + "&lon=" + String(longitude) + "&units=metric&appid=" + String(API\_KEY);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +

               "Host: " + host + "\r\n" +

               "Connection: close\r\n\r\n");

  // Chờ đến khi nhận được phản hồi từ API

  while (!client.available()) {

    delay(1000);

    Serial.println("Đang đợi phản hồi");

  }

// Đọc dữ liệu từ phản hồi và lưu vào một String

String response = "";

while (client.available()) {

response += client.readString();

}

// Phân tích dữ liệu JSON từ phản hồi

const size\_t capacity = JSON\_OBJECT\_SIZE(4) + 120;

DynamicJsonDocument doc(capacity);

deserializeJson(doc, response);

// Lấy các giá trị dự báo thời tiết từ JSON

const char\* weatherDescription = doc["weather"][0]["description"];

float temperature = doc["main"]["temp"];

int humidity = doc["main"]["humidity"];

int pressure = doc["main"]["pressure"];

// Hiển thị thông tin dự báo thời tiết trên màn hình TFT

  tft.setBackgroundColor(COLOR\_BLACK);

  tft.setFont(Terminal6x8);

  tft.drawText(0, 0, "Du bao thoi tiet hom nay:", COLOR\_RED);

  tft.drawText(0, 30, "HOM NAY TOI BUOI", COLOR\_YELLOW);

    tft.setFont(Terminal6x8);

    tft.drawText(0, 60, "Temperature: " + String(temperature) + " C", COLOR\_BLUE);

    tft.drawText(0, 90, "Humidity: " + String(humidity) + "%", COLOR\_WHITE);

    tft.drawText(0, 120, "Pressure: " + String(pressure) + " hPa", COLOR\_GREEN);

 //   tft.drawText(0, 150, "description: " + weatherDescription, COLOR\_BLUE);

    tft.drawText(0, 180, "DM HUY BIP", COLOR\_ORANGE);

  // Chờ 5 phút trước khi lấy dữ liệu dự báo thời tiết mới

  delay(300000);

}

ARDUINO WITH ds1307

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include "RTClib.h"

RTC\_DS1307 rtc;

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

void setup ()

{

 Serial.begin(9600);

 if (! rtc.begin())

 {

   Serial.print("Couldn't find RTC");

   while (1);

 }

  if (! rtc.isrunning())

 {

   Serial.print("RTC is NOT running!");

   Serial.println();

 }

   rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

   //rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));

}

void loop ()

{

   DateTime now = rtc.now();

   if(now.hour()<=9)

   {

     Serial.print("0");

     Serial.print(now.hour());

   }

   else {

    Serial.print(now.hour());

   }

   Serial.print(':');

   if(now.minute()<=9)

   {

     Serial.print("0");

     Serial.print(now.minute());

   }

   else {

    Serial.print(now.minute());

   }

   Serial.print(':');

   if(now.second()<=9)

   {

     Serial.print("0");

     Serial.print(now.second());

   }

   else {

    Serial.print(now.second());

   }

   Serial.println();

   Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);

   Serial.print(",");

   if(now.day()<=9)

   {

     Serial.print("0");

     Serial.print(now.day());

   }

   else {

    Serial.print(now.day());

   }

   Serial.print('/');

   if(now.month()<=9)

   {

     Serial.print("0");

     Serial.print(now.month());

   }

   else {

    Serial.print(now.month());

   }

   Serial.print('/');

   if(now.year()<=9)

   {

     Serial.print("0");

     Serial.print(now.year());

   }

   else {

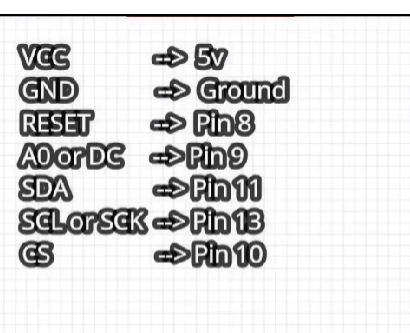
    Serial.print(now.year());

   }

   Serial.println();

   delay(1000);

}



1307 va tft 1.8”

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_ST7735.h>

#include "RTClib.h"

RTC\_DS1307 rtc;

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

// Pin definitions for the TFT display

#define TFT\_CS 10

#define TFT\_RST 8

#define TFT\_DC 9

// Create an instance of the TFT display

Adafruit\_ST7735 tft = Adafruit\_ST7735(TFT\_CS, TFT\_DC, TFT\_RST);

void setup ()

{

  Serial.begin(9600);

  // Initialize the TFT display

  tft.initR(INITR\_BLACKTAB);

  tft.fillScreen(ST7735\_BLACK);

  if (! rtc.begin())

  {

    Serial.print("Couldn't find RTC");

    while (1);

  }

  if (! rtc.isrunning())

  {

    Serial.print("RTC is NOT running!");

    Serial.println();

  }

  rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

}

void loop ()

{

  DateTime now = rtc.now();

  tft.setCursor(0, 0);

  tft.setTextColor(ST7735\_WHITE);

  tft.setTextSize(2);

  if(now.hour()<=9)

  {

    tft.print("0");

    tft.print(now.hour());

  }

  else {

    tft.print(now.hour());

  }

  tft.print(':');

  if(now.minute()<=9)

  {

    tft.print("0");

    tft.print(now.minute());

  }

  else {

    tft.print(now.minute());

  }

  tft.print(':');

  if(now.second()<=9)

  {

    tft.print("0");

    tft.print(now.second());

  }

  else {

    tft.print(now.second());

  }

  tft.setCursor(0, 25);

  tft.setTextColor(ST7735\_GREEN);

  tft.setTextSize(1);

  tft.print(daysOfTheWeek[now.dayOfTheWeek()]);

  tft.print(",");

  if(now.day()<=9)

  {

    tft.print("0");

    tft.print(now.day());

  }

  else {

    tft.print(now.day());

  }

  tft.print('/');

  if(now.month()<=9)

  {

    tft.print("0");

    tft.print(now.month());

  }

  else {

    tft.print(now.month());

  }

  tft.print('/');

  if(now.year()<=9)

  {

    tft.print("0");

    tft.print(now.year());

  }

  else {

    tft.print(now.year());

  }

  delay(1000);

}

Buzzer  
#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include "RTClib.h"

RTC\_DS1307 rtc;

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

// Khởi tạo đối tượng LiquidCrystal\_I2C với địa chỉ của màn hình là 0x27

LiquidCrystal\_I2C lcd(0x27, 16, 2);

const int buttonMenuPin = 2; // Nút 1 - Mở menu đặt báo thức

const int buttonHourPin = 3; // Nút 2 - Set giờ

const int buttonMinutePin = 4; // Nút 3 - Set phút

const int buzzerPin = 5; // Chân điều khiển buzzer

int buttonMenuState = HIGH; // Trạng thái của nút 1

int buttonHourState = HIGH; // Trạng thái của nút 2

int buttonMinuteState = HIGH; // Trạng thái của nút 3

int lastButtonMenuState = HIGH; // Trạng thái trước đó của nút 1

int lastButtonHourState = HIGH; // Trạng thái trước đó của nút 2

int lastButtonMinuteState = HIGH; // Trạng thái trước đó của nút 3

int menuLevel = 0; // Mức menu đang mở (0 = không mở, 1 = đặt giờ, 2 = đặt phút)

int alarmHour = 0; // Giờ đặt báo thức

int alarmMinute = 0; // Phút đặt báo thức

bool alarmOn = false; // Báo thức có đang được bật hay không

void setup() {

  Serial.begin(9600);

  lcd.init();  // Khởi tạo màn hình

  lcd.backlight();  // Bật đèn nền màn hình

  Wire.begin();

  pinMode(buttonMenuPin, INPUT\_PULLUP); // Khai báo chân nút 1 là INPUT\_PULLUP

  pinMode(buttonHourPin, INPUT\_PULLUP); // Khai báo chân nút 2 là INPUT\_PULLUP

  pinMode(buttonMinutePin, INPUT\_PULLUP); // Khai báo chân nút 3 là INPUT\_PULLUP

  pinMode(buzzerPin, OUTPUT); // Khai báo chân điều khiển buzzer là OUTPUT

  if (!rtc.begin()) {

    Serial.println("Couldn't find RTC");

    while (1);

  }

  if (!rtc.isrunning()) {

    Serial.println("RTC is NOT running!");

    Serial.println();

  }

  rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

}

void loop(){

  DateTime now = rtc.now();

  // Xóa màn hình

  lcd.clear();

  // Hiển thị giờ phút giây

  lcd.setCursor(0, 0);

  if (now.hour() <= 9) {

    lcd.print("0");

  }

  lcd.print(now.hour());

  lcd.print(':');

  if (now.minute() <= 9) {

    lcd.print("0");

  }

  lcd.print(now.minute());

  lcd.print(':');

  if (now.second() <= 9) {

    lcd.print("0");

  }

  lcd.print(now.second());

  // Hiển thị ngày tháng năm

  lcd.setCursor(0, 1);

  lcd.print(daysOfTheWeek[now.dayOfTheWeek()]);

  lcd.print(",");

  if (now.day() <= 9) {

    lcd.print("0");

  }

  lcd.print(now.day());

  lcd.print('/');

  if (now.month() <= 9) {

    lcd.print("0");

  }

  lcd.print(now.month());

  lcd.print('/');

  lcd.print(now.year());

switch (menuLevel) {

case 0: // Chế độ bình thường

buttonMenuState = digitalRead(buttonMenuPin);

if (buttonMenuState != lastButtonMenuState) {

if (buttonMenuState == LOW) {

// Nếu nút 1 được nhấn, mở menu đặt báo thức

menuLevel = 1;

}

}

lastButtonMenuState = buttonMenuState;

break;

case 1: // Đặt giờ báo thức

  buttonHourState = digitalRead(buttonHourPin);

  if (buttonHourState != lastButtonHourState) {

    if (buttonHourState == LOW) {

      // Nếu nút 2 được nhấn, tăng giờ báo thức

      alarmHour++;

      if (alarmHour > 23) {

        alarmHour = 0;

      }

      lcd.clear();

      lcd.print("Set Alarm:");

      lcd.setCursor(0, 1);

      lcd.print(alarmHour);

      lcd.print(":");

      lcd.print(alarmMinute);

    }

  }

  lastButtonHourState = buttonHourState;

  buttonMinuteState = digitalRead(buttonMinutePin);

  if (buttonMinuteState != lastButtonMinuteState) {

    if (buttonMinuteState == LOW) {

      // Nếu nút 3 được nhấn, tăng phút báo thức

      alarmMinute++;

      if (alarmMinute > 59) {

        alarmMinute = 0;

      }

      lcd.clear();

      lcd.print("Set Alarm:");

      lcd.setCursor(0, 1);

      lcd.print(alarmHour);

      lcd.print(":");

      lcd.print(alarmMinute);

    }

  }

  lastButtonMinuteState = buttonMinuteState;

  break;

}

// Kiểm tra báo thức

if (now.hour() == alarmHour && now.minute() == alarmMinute && now.second() == 0 && !alarmOn) {

alarmOn = true;

lcd.clear();

lcd.print("Alarm ON ");

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

delay(500);

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

delay(500);

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

delay(500);

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

delay(500);

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

delay(500);

} else if (alarmOn && now.second() == 30) {

// Khi đồng hồ chạy đến giây thứ 30 của phút, tắt báo thức

alarmOn = false;

lcd.clear();

lcd.print(daysOfTheWeek[now.dayOfTheWeek()]);

lcd.print(",");

if (now.day() <= 9) {

lcd.print("0");

}

lcd.print(now.day());

lcd.print('/');

if (now.month() <= 9) {

lcd.print("0");

}

lcd.print(now.month());

lcd.print('/');

lcd.print(now.year());

}

delay(1000);

}