# BÁO CÁO PHẦN HIỂN THỊ MÀN HÌNH TFT-NCKH ĐỢT 1 (THOA)

## I. Lý thuyết:

**Mục tiêu**: hiển thị giao diện nền màn hình, chuyển giao diện màn hình, tắt mở màn hình và kèm theo hiện đồng hồ thời gian thực.

## II. Báo cáo tiến trình thực hiện:

### 2.1 Thời gian thực cho đồng hồ:

**Mục tiêu:** Hiển thị thời gian thực (giờ:phút:giây) từ Internet lên màn hình TFT tròn GC9A01.

**Cách thực hiện:**

- Sử dụng kết nối Wi-Fi để đồng bộ thời gian từ máy chủ NTP (pool.ntp.org).

- Sau khi Wi-Fi kết nối thành công, lệnh configTime(...) được dùng để thiết lập múi giờ GMT+7 (Việt Nam).

- Hàm getLocalTime() lấy thông tin thời gian hiện tại.

- Thời gian được định dạng bằng strftime() và in lên màn hình qua tft.println(...).

**Hạn chế**: Khi mất mạng thì thời gian sẽ không được cập nhật mới.

**Code và giải thích:**

const char\* ssid     = "BenTre2004"; // phần này là nhập tên Wifi

const char\* password = "thothomphuc"; // nhập mật khẩu Wifi

// NTP thiết lập múi giờ Việt Nam +7

const char\* ntpServer = "pool.ntp.org";

const long gmtOffset\_sec = 7 \* 3600;

const int daylightOffset\_sec = 0;

// Kiểm tra Wifi có được kết nối chưa

WiFi.begin(ssid, password);

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

    delay(500);

    Serial.print(".");

  }

  Serial.println("\nWiFi connected!");

  tft.setCursor(60, 120);

  tft.println("Xin Chao");

  delay(5000);

  // Đồng bộ thời gian từ NTP

  configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

// hàm gọi thời gian thực

void printLocalTime() {

  struct tm timeinfo;

  if (!getLocalTime(&timeinfo)) {

    tft.setCursor(60, 120);

    tft.println("Time error"); // kết nối thất bại

    return;

  }

  char buffer[30];

  strftime(buffer, sizeof(buffer), "%A, %d-%m-%Y", &timeinfo); // Định nghĩa thứ ngày tháng và năm

  char timeStr[10];

  strftime(timeStr, sizeof(timeStr), "%H:%M:%S", &timeinfo); // Định nghĩa giờ phút và giây

  tft.setCursor(20, 100);

  tft.println(buffer); // hiện thứ ngày tháng năm ra màn hình

  tft.setCursor(60, 120);

  tft.println(timeStr); // hiện giờ phút giây ra màn hình

}

### 2.2 Chuyển giao diện màn hình:

**Mục tiêu**: Cho phép chuyển qua 3 giao diện hiển thị khác nhau bằng nút nhấn.

**Cách thực hiện:**

- Nút nhấn số 1 gắn vào **GPIO 12**, khai báo là INPUT\_PULLUP (không cần điện trở ngoài).

- Biến screenIndex dùng để theo dõi giao diện hiện tại.

- Khi nút được nhấn, biến screenIndex tăng lên và drawScreen(screenIndex) được gọi để cập nhật màn hình.

- Lệnh tft.fillScreen(...) hoặc tft.pushImage(...) được dùng để thay đổi nội dung hiển thị.

**Code và giải thích:**

const int buttonNext = 12;  // nút 1: chuyển màn hình

bool lastNextState = HIGH; // ở mức cao ( chưa nhấn)

pinMode(buttonNext, INPUT\_PULLUP); // nút nhấn ở dạng PULLUP

// phần nằm trong hàm loop

 // Nút 1: Chuyển giao diện

  bool nextState = digitalRead(buttonNext);

  if (nextState == LOW && lastNextState == HIGH) {

    delay(100);  // chống dội

    screenIndex++;

    if (screenIndex > 2) screenIndex = 0;

    if (screenOn) drawScreen(screenIndex);

  }

  lastNextState = nextState;

// hàm gọi các giao diện để chuyển

void drawScreen(int index) {

  if (index == 0) {

// giao diện 1

    tft.fillScreen(TFT\_BLACK);

    tft.pushImage(0, 0, 240, 240, vutru\_bitmap\_vutru);

    tft.fillRect(10, 95, 200, 40, TFT\_BLACK);

    printLocalTime();

    delay(300);

  } else if (index == 1) {

// giao diện 2

    tft.fillScreen(TFT\_BLUE);

    tft.setCursor(60, 100);

    tft.setTextColor(TFT\_WHITE);

    tft.setTextSize(3);

    tft.println("HELLO");

  } else if (index == 2) {

// giao diện 3

    tft.fillScreen(TFT\_WHITE);

    tft.setCursor(40, 100);

    tft.setTextColor(TFT\_BLACK);

    tft.setTextSize(3);

    tft.println("GOODBYE");

  }

}

### 2.3 Tắt mở màn hình:

**Mục tiêu:** Tiết kiệm năng lượng khi không cần sử dụng màn hình.

**Cách thực hiện:**

- Nút nhấn số 2 gắn vào **GPIO 14**, cũng dùng INPUT\_PULLUP.

- Khi nút được nhấn, biến screenOn chuyển đổi giữa true và false.

- Lệnh tft.writecommand(TFT\_DISPOFF) được dùng để **tắt** màn hình.

- Lệnh tft.writecommand(TFT\_DISPON) để **bật** màn hình lại.

- Nếu màn hình được bật lại, hệ thống tự động vẽ lại giao diện tương ứng với screenIndex.

**Code và giải thích:**

const int buttonPower = 14; // nút 2: bật/tắt màn hình

bool lastPowerState = HIGH; // ở mức cao ( chưa nhấn)

pinMode(buttonPower, INPUT\_PULLUP); // ở chế độ PULLUP

// hàm loop

 // Nút 2: Bật / tắt màn hình

  bool powerState = digitalRead(buttonPower);

  if (powerState == LOW && lastPowerState == HIGH) {

    delay(100);  // chống dội

    screenOn = !screenOn;

    if (screenOn) {

      tft.writecommand(TFT\_DISPON);  // Bật lại

      drawScreen(screenIndex);

    } else {

      tft.writecommand(TFT\_DISPOFF); // Tắt

    }

  }

  lastPowerState = powerState;

### 2.4 Hiển thị nền màn hình:

**Mục tiêu:** Làm rõ nội dung bằng cách hiển thị nền tương phản phù hợp với từng giao diện.

**Cách thực hiện:**

* Giao diện 1: dùng ảnh bitmap (tft.pushImage(...)) phủ kín màn hình 240x240.
* Giao diện 2: nền xanh (tft.fillScreen(TFT\_BLUE)), chữ trắng “HELLO”.
* Giao diện 3: nền trắng (tft.fillScreen(TFT\_WHITE)), chữ đen “GOODBYE”.

**Note:** phần nội dung chỉ test để demo nên chưa hoàn thiện

**Code và giải thích:**

// phần file .h với tên là vutru dạng bitmap để vẽ ảnh nền

#ifndef VUTRU\_H

#define VUTRU\_H

const uint16\_t vutru\_bitmap\_vutru [] PROGMEM = {

  0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,

  0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,

  0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0821, 0x0020, 0x0000, 0x0000, 0x0821, 0x0000, 0x736e,... };

const int vutru\_bitmap\_allArray\_LEN = 1;

const uint16\_t\* vutru\_bitmap\_allArray[1] = {

  vutru\_bitmap\_vutru

};

#endif

// hàm chính sử dụng bằng cách khai báo và gọi hàm

#include "vutru.h"

tft.pushImage(0, 0, 240, 240, vutru\_bitmap\_vutru);

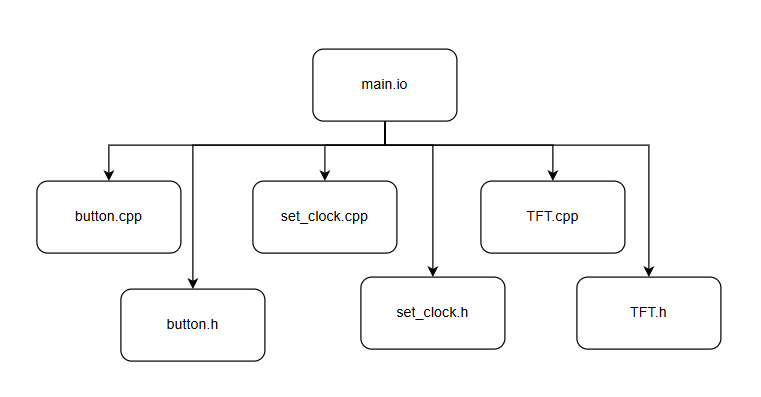
# BÁO CÁO PHẦN HIỂN THỊ MÀN HÌNH TFT-NCKH ĐỢT 2

## I Lý thuyết:

**Mục tiêu**: Có thay đổi về phần thời gian ( đổi từ lấy thời gian thông qua NTP của wifi thì chuyển sang sử dụng module RTC DS1302). Đã chia module thành từng phần khác nhau để dễ quản lý.

## II. Báo cáo tiến trình thực hiện:

### 2.1 Sơ đồ module thực hiện:



### 2.2 Thời gian thực cho đồng hồ:

**Mục tiêu**: Chạy theo thời gian thực mà không cần wifi.

**Tiến trình:** Hiện tại code chạy trên Serial của app Arduino IDE thì nó hiện giờ, còn khi code cho TFT thì nó không nhận định được thời gian và báo “ Unknown day 2000-00-00 00:00:00”

**Code:**

* **File .cpp:**

#include <TFT\_eSPI.h>

#include <SPI.h>

#include <stdio.h>

#include <DS1302.h>

#include "set\_clock.h"

#include "button.h"

#include "TFT.h"

const int kCePin   = 19;  // Chip Enable

const int kIoPin   = 22;  // Input/Output

const int kSclkPin = 21;  // Serial Clock

extern TFT\_eSPI tft;

// Create a DS1302 object.

DS1302 rtc(kCePin, kIoPin, kSclkPin);

String dayAsString(const Time::Day day) {

  switch (day) {

    case Time::kSunday: return "Sunday";

    case Time::kMonday: return "Monday";

    case Time::kTuesday: return "Tuesday";

    case Time::kWednesday: return "Wednesday";

    case Time::kThursday: return "Thursday";

    case Time::kFriday: return "Friday";

    case Time::kSaturday: return "Saturday";

  }

  return "(unknown day)";

}

void printTime() {

  Time t = rtc.time();

  String day = dayAsString(t.day);

  char buf[50];

  snprintf(buf, sizeof(buf), "%s %04d-%02d-%02d %02d:%02d:%02d",

           day.c\_str(), t.yr, t.mon, t.date, t.hr, t.min, t.sec);

  tft.setTextColor(TFT\_WHITE);

  tft.fillScreen(TFT\_BLACK);

  tft.setTextSize(2);

  tft.setCursor(50, 100);

  tft.println(buf);

}

void rtcInit() {

  rtc.writeProtect(false);

  rtc.halt(false);

  // Cài đặt thời gian mặc định (có thể sửa theo nhu cầu)

  Time t(2025, 8, 2, 15, 4, 30, Time::kSaturday);

  rtc.time(t);

}

* **File .h**

#ifndef SET\_CLOCK\_H

#define SET\_CLOCK\_H

#include <stdint.h>

void rtcInit();       //khoi tao thoi gian

void printTime();

#endif

### 2.3 Nút nhấn:

**Tiến trình:** Nút nhấn chạy ổn định.

**Code:**

* **File .cpp**

#include <TFT\_eSPI.h>

#include <SPI.h>

#include <stdio.h>

#include <DS1302.h>

#include "set\_clock.h"

#include "button.h"

#include "TFT.h"

const int buttonNext = 12;  // nút 1: chuyển màn hình

const int buttonPower = 14; // nút 2: bật/tắt màn hình

bool lastNextState = HIGH;

bool lastPowerState = HIGH;

extern TFT\_eSPI tft;

extern int screenIndex;

extern bool screenOn;

void buttonInit() {

  pinMode(buttonNext, INPUT\_PULLUP);

  pinMode(buttonPower, INPUT\_PULLUP);

}

//hàm nhấn nút

void handleButtons() {

  // Nút chuyển giao diện

  bool nextState = digitalRead(buttonNext);

  if (nextState == LOW && lastNextState == HIGH) {

    delay(100); // chống dội

    screenIndex = (screenIndex + 1) % 3;

    if (screenOn) drawScreen(screenIndex);

  }

  lastNextState = nextState;

// Nút bật/tắt màn hình

  bool powerState = digitalRead(buttonPower);

  if (powerState == LOW && lastPowerState == HIGH) {

    delay(100); // chống dội

    screenOn = !screenOn;

    if (screenOn) {

      tft.writecommand(TFT\_DISPON);

      drawScreen(screenIndex);

    } else {

      tft.writecommand(TFT\_DISPOFF);

    }

  }

  lastPowerState = powerState;

}

* **File .h**

#ifndef BUTTON\_H

#define BUTTON\_H

void buttonInit();

void handleButtons();

#endif

### 2.4 Hiện ra màn hình

**Tiến trình:** Màn hình chuyển mượt. Đang tìm hiểu thêm cách vẽ nền màn hình cho có thẩm mỹ.

**Code:**

* **File .cpp**

#include <TFT\_eSPI.h>

#include <SPI.h>

#include <stdio.h>

#include <DS1302.h>

#include "set\_clock.h"

#include "button.h"

#include "TFT.h"

TFT\_eSPI tft = TFT\_eSPI();

// Biến toàn cục để quản lý trạng thái

int screenIndex = 0;

bool screenOn = true;

void tftInit() {

  tft.init();

  tft.setRotation(0);

}

void drawScreen(int index) {

  if (index == 0) {

    image1 ();

  } else if (index == 1) {

    image2();

  } else if (index == 2) {

    image3();

  }

}

void image1 () {

  printTime();

  delay(300);

}

void image2 () {

  tft.fillScreen(TFT\_BLUE);

  tft.setCursor(60, 100);

  tft.setTextColor(TFT\_WHITE);

  tft.setTextSize(3);

  tft.println("HELLO");

}

void image3 () {

  tft.fillScreen(TFT\_WHITE);

  tft.setCursor(40, 100);

  tft.setTextColor(TFT\_BLACK);

  tft.setTextSize(3);

  tft.println("GOODBYE");

}

* **File .h**

#ifndef TFT\_H

#define TFT\_H

extern int screenIndex;

extern bool screenOn;

void tftInit();

void drawScreen(int index);

void image1 ();

void image2 ();

void image3 ();

#endif

### 2.5 File main để chạy code:

#include <TFT\_eSPI.h>

#include <SPI.h>

#include <stdio.h>

#include <DS1302.h>

#include "set\_clock.h"

#include "button.h"

#include "TFT.h"

void setup() {

  Serial.begin(115200);

  tftInit();

  buttonInit();

  rtcInit();

  drawScreen(screenIndex);

}

void loop() {

  handleButtons();

}