



IOT102_BÁO CÁO Topic 12_NHÓM 5

🕒 Created	@January 27, 2022 9:14 AM
☰ Môn học	IOT102
☰ Lớp	SE1608
☰ Nhóm	5
☰ Thành viên nhóm 1	Nguyễn Xuân Duy - SE160895
☰ Thành viên nhóm 2	Vũ An Ninh - SE160193
☰ Thành viên nhóm 3	Phạm Trung Nguyên - SE160896

0. Mục lục

0. Mục lục

I. Giới thiệu

0. Link các thành phần dự án
1. Mô tả dự án
2. Ứng dụng thực tế (ứng dụng, cách sử dụng, ưu điểm, hạn chế)

II. Phần cứng (Hardware)

1. **Thành phần**
2. Properties of components
3. Schematic design

III. Phần mềm (Software)

1. Giải thích source code
2. Flowchart
3. Source code

IV. Lắp đặt và chạy thử

1. Thiết kế (hướng dẫn lắp đặt)
2. Lỗi phát sinh có thể có
3. Các bộ thử
4. Mong đợi

V. Tổng kết

VI. Phụ lục

1. Link các thành phần khác

I. Giới thiệu

0. Link các thành phần dự án

- Slide thuyết trình dự án: <https://drive.google.com/file/d/1E-vpunbNvQATjfHIQpBGdAqpwXs6wENZ/view?usp=sharing>
- Link folder Google Drive **chứa đầy đủ file** của dự án: https://drive.google.com/drive/folders/1IBHVpx1-_fq4CDDDuwYF6XdEJCpXtOeD?usp=sharing
- Link file **Google Docs** chứa **code Arduino**: https://docs.google.com/document/d/1UYSCy_0L3gpxe4USolQZER3TSKp0voxbsf2Wdkk8vNY/edit?usp=sharing
- Ảnh chụp màn hình của **mạch mô phỏng** trong Proteus: https://docs.google.com/document/d/1E45J7bOGvAFHA3lzE5OuceteYBaYdNOTX_wkIlAatE8/edit?usp=sharing

1. Mô tả dự án

a) Tên dự án: Hệ thống chuông tự động sử dụng trong trường học.

b) Yêu cầu:

- Lắp đặt sử dụng thời gian thực.
- Có thể điều chỉnh số lượng tiết học (slots) mỗi ngày (số lần chuông reng mỗi ngày).
- Hiển thị các thông tin: hiện tại đang là slot hoặc giờ giải lao (break) thứ mấy, hiển thị thời gian hết slot, thời gian hết giờ giải lao.

2. Ứng dụng thực tế (ứng dụng, cách sử dụng, ưu điểm, hạn chế)

a) Ứng dụng:

- Là thiết bị được thiết lập để có thể lấy thời gian thực một cách chuẩn xác và tự động reng chuông dựa trên những thông tin được lập trình từ trước bởi người sử dụng, quản lí.
- Thay thế các thao tác bấm chuông thủ công. Tuy nhiên vẫn có nút bấm để cho chuông reng bất kỳ lúc nào.
- Sử dụng trong các trường học, cơ sở giáo dục tư nhân, trường đại học, hoặc thậm chí nhà xưởng, cơ sở sản xuất,... Được lập trình cài đặt mặc định là sử dụng cho trường Đại học FPT TP. HCM.

b) Mô tả khái quát cách sử dụng:

- Ban đầu khi bật hệ thống lên (hoặc tắt đi bật lại), vì chưa cài đặt thông tin về các tiết học nên hệ thống sẽ chỉ hiển thị thời gian hiện tại với bố cục: ngày/tháng/năm, giờ:phút:giây

- Để thiết lập thời gian reng chuông, ta nhấn chọn phím '*' bất kỳ lúc nào trên keypad khi vừa khởi động thiết bị, vừa reset hoặc đang trong lúc OLED hiện thông tin các slot.
- Khi đó màn hình OLED sẽ hiện lên một menu gồm 3 lựa chọn:
 1. **Set default slot:** Ở đây hệ thống sẽ thiết lập mặc định một ngày sẽ có 6 tiết học, mỗi tiết học kéo dài 1 giờ 30 phút. Thời gian giải lao giữa mỗi tiết học là 15 phút (short break), thời gian giải lao giữa tiết học thứ 3 và thứ 4 là 30 phút (long break). Lựa chọn mặc định này được áp dụng thực tế cho thời gian biểu mỗi ngày 6 slots của trường Đại học FPT TP. HCM.
 2. **Custom your slot:** Khi chọn lựa chọn này, người dùng sẽ được đưa vào một giao diện để nhập thời gian của một tiết học, số tiết, thời gian nghỉ,... Cụ thể:
 - Nhập số slot có trong ngày, hoặc số slot còn lại tính từ thời điểm bắt đầu nhập:
 - Nếu nhập trước 7:00 thì nhập số slot có hôm đó.
 - Nếu nhập sau 7:00 thì nhập số slot còn lại trong ngày tính từ thời điểm nhập.
 - Nếu nhập sau 17:30 thì nhập số slot của ngày hôm sau.
 - Nhập thời lượng 1 slot, thời lượng giải lao ngắn (giải lao sau mỗi tiết học), thời lượng giải lao dài (tính bằng phút). Sau slot bao nhiêu thì đến thời gian giải lao dài.
 - Sau khi nhập đầy đủ thông tin, thiết bị sẽ kiểm tra tổng thời gian của tất cả các slot học, thời gian nghỉ có vượt quá thời điểm cuối ngày (17:30) không, nếu có thì yêu cầu nhập lại hoặc thoát, nếu không thì sử dụng các số liệu được nhập và tiếp tục.
 - Nếu sử dụng lựa chọn này, số thứ tự slot hiện tại sẽ được tính theo số liệu được nhập, và slot 1 được mặc định là bắt đầu vào lúc 7:00. Mặc định thời điểm kết thúc slot cuối cùng là 17:30.
 3. **Exit:** thoát khỏi menu và quay trở lại màn hình chính
- Người dùng nhấn chọn các số từ 1 → 3 khi menu hiện lên, nếu nhập sai sẽ bị yêu cầu nhập lại.
- Sau khi nhập xong tất cả lựa chọn, màn hình OLED sẽ căn cứ vào thông tin được nhập và hiện lên: thời gian hiện tại, hiện tại là slot mấy hoặc giờ giải lao của slot mấy, thời điểm hết slot hoặc hết giờ giải lao.
- Đến đúng giờ, chuông sẽ tự reng. Ngoài ra còn có một nút bấm để cho chuông reng bất kỳ lúc nào.

c) Ưu điểm:

- Tự động hóa các thao tác cho chuông reng, chỉ cần nhập thông tin một lần và sử dụng được trong nhiều ngày (nếu không tắt thiết bị).
- Chính xác, có độ tin tưởng cao.
- Lựa chọn mặc định được thêm vào làm việc áp dụng thiết bị cho trường Đại học FPT TP. HCM trở nên dễ dàng. Ngoài ra lựa chọn tùy chỉnh, người dùng có thể tùy chỉnh tất cả các chức năng.

d) Hạn chế:

- Hạn chế về khả năng mô phỏng khiến thời gian thực khi chạy mô phỏng đôi khi lệch với thời gian thực tế. Khả năng mô phỏng phụ thuộc vào laptop.

- Chưa có thao tác xóa khi nhập dữ liệu vì hạn chế của keypad.
- Màn hình OLED hiển thị còn hạn chế.

II. Phần cứng (Hardware)

1. Thành phần

>>> **Lưu ý:** phần ngoặc tròn (...) chỉ định tên của thiết bị trong phần mềm mô phỏng Proteus.

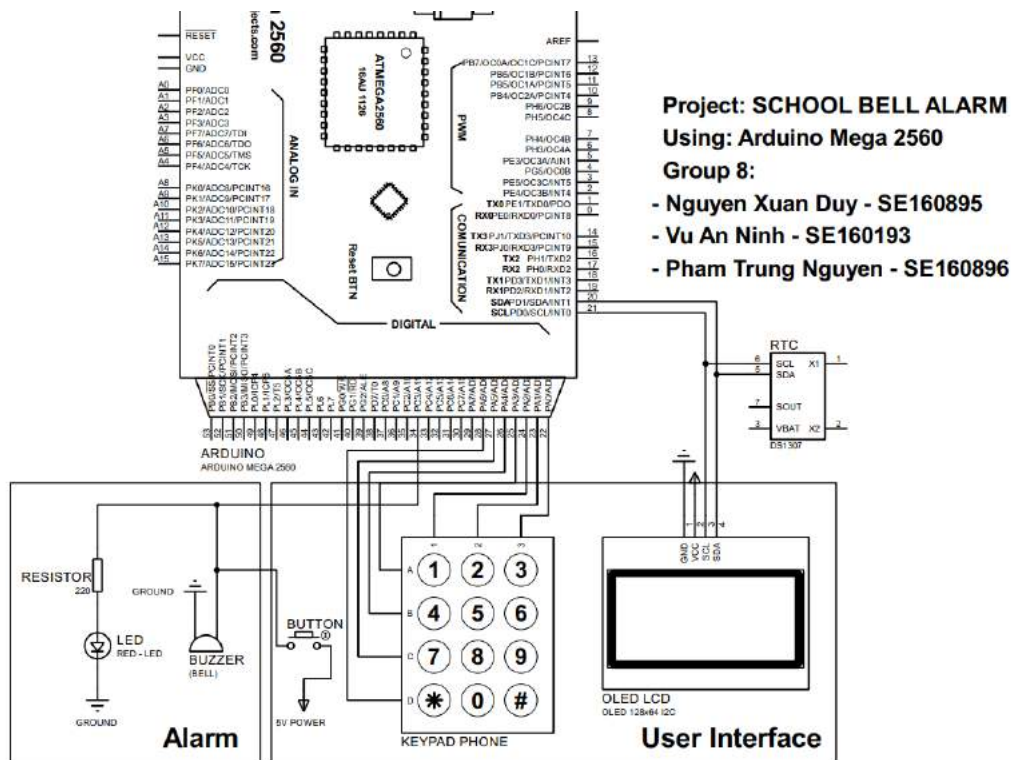
- 01 Arduino Mega 2560
- 01 Push Button (BUTTON): Khi nhấn nút chuông sẽ bắt đầu reo
- 01 Buzzer (BUZZER): chuông
- 01 Real-time Clock DS1307 (DS1307): Thiết bị dùng để lấy thời gian thực
- 01 LCD OLED 128x64 **tích hợp I2C** (OLED12864I2C) : dùng để hiển thị thông tin của thời gian hiện tại và các thông tin tương tác người dùng (User Interface). Các thông tin hiển thị bao gồm:
 - ngày/tháng/năm
 - Giờ/phút/giây
 - Thông tin slot hiện tại: slot mấy, thời gian kết thúc slot
 - Các menu, lựa chọn, nhập input,... dành cho người dùng.
- 01 đèn LED (LED-RED):
 - Đèn sẽ được bật sáng khi chuông reo
- 01 Keypad dành cho điện thoại (KEYPAD-PHONE): Dùng để lấy dữ liệu người dùng nhập vào.
- 01 điện trở (RESISTOR)
- Dây dẫn, nguồn điện (POWER), ground (GROUND) (cực âm, nối đất)

2. Properties of components

- RESISTOR: 220 Ω
- BUZZER:
 - Operating Voltage: 5V
 - Load Resistance: 100
 - Frequency: 500Hz
- OLED LCD:
 - 128x64 pixels
 - Có tích hợp sẵn I2C
- POWER: 5V
- DS1307:

- PCB Package: DIL08
- Button:
 - Off Resistance: 100M
 - On Resistance: 100m
 - Switching Time: 1m

3. Schematic design



III. Phần mềm (Software)

1. Giải thích source code

a) Các thư viện được sử dụng:

>>> Link file .ZIP của các thư viện được đính kèm ở phần đầu báo cáo.

- Các thư viện dùng cho việc set up, sử dụng và điều khiển màn hình OLED tích hợp I2C (OLED128612C):
 - `Adafruit_SSD1306.h`
 - `splash.h`

- `Adafruit_GFX.h`
- `Adafruit_SPITFT.h`
- `Adafruit_SPITFT_Macros.h`
- `Wire.h`
- Thư viện dùng để thao tác, sử dụng Keypad-phone:
 - `Keypad.h`
- Các thư viện khác:
 - `string.h` : chứa các thao tác xử lý xâu, chuỗi.
- Thư viện dùng để lấy thông tin thời gian thực từ Real-time Clock DS1307:
 - `RTCLib.h`

b) Giải thích các biến global được sử dụng:

- Các biến dành cho thao tác keypad:
 - `ROWS, COLS` : số hàng, số cột của keypad
 - `keys[ROWS][COLS]` : mảng 2 chiều dùng thao tác keypad
 - `rowPins[ROWS], colPins[COLS]` : khai báo chân cắm của các hàng và cột
 - `keypad` : biến dùng như đối tượng để thao tác keypad
- `numSlot` : Tổng số lượng slot có trong ngày
- `currentSlot` : Slot hiện tại
- `noSlotB4LongBreak` : Tổng số slot trước long break
- `shortBreak, longBreak` : thời gian nghỉ ngắn, nghỉ dài
- `startTime, endTime` : thời gian bắt đầu ngày mới (bắt đầu slot 1, tức 7:00), và kết thúc ngày (kết thúc slot cuối, tức 17:30), thay đổi được.
- `slotInfo[100][2]` : `[i][]` : là slot thứ i, `[i][0]` là slot thứ i bắt đầu lúc nào, `[i][1]` là slot thứ i kết thúc lúc nào, thang đo tính bằng giây, đánh số từ slot 1 là `[1][0]`
- `TIME_BELL_RING, bellPosition` : thời gian chuông reng, chân cắm của chuông.
- `rtc` : biến dùng như đối tượng để thao tác real-time clock

c) Khái quát các hàm trong source code:

- `setup()` : dùng để set up các hàm, thiết bị, bắt đầu chương trình Arduino (hàm mặc định trong code của Arduino):
 - Bắt đầu đối tượng `rtc` : bắt đầu hàm lấy giá trị thời gian thực.
 - Khởi tạo đối tượng `display` : dùng để khởi tạo đối tượng điều khiển màn hình OLED OLED 128x64 tích hợp I2C.
 - Khởi tạo các thông số bắt đầu (`setCursor, setTextColor,...`).

- `int getKeypad(msg)` : In `msg` ra màn hình OLED (là câu hỏi input, mặc định là “Enter number: “), yêu cầu người dùng nhập số từ keypad. Người dùng bấm số nào từ keypad thì số đó ra màn hình OLED:
 - Biến `str` là biến local (kiểu `String`) dùng để lưu giá trị được nhập vào dưới dạng chuỗi.
 - Nếu giá trị được nhấn là số thì thêm số đó vào phía sau `str` và hiện ra trên màn hình. Vì mặc định giá trị được truyền vào `inputKey` khi nhập từ Keypad là kiểu `byte`, nên cần chuyển `inputKey` sang `String`, sau đó gán vào sau chuỗi `str` bằng cách cộng chuỗi.
 - Người dùng nhập số từ keypad đến khi nào phím được nhấn là # hoặc * thì dừng. 2 phím này không được tính vào input. Giá trị input hiện tại có sẵn trong `str` được chuyển từ `String` về kiểu `int` bằng hàm `toInt()` và `return` vào giá trị của hàm `getKeypad()`.
- `ringBell()` : Dùng để rung chuông trong khoảng thời gian `TIME_BELL_RING` (hiện đang được mặc định là 2 giây).
- `turnOnOLED(msg)` : Dùng để in các dòng `msg` ra màn hình OLED.
- `turnOffOLED()` : tắt màn hình OLED, ngưng hiển thị các dòng thông báo.
- `getCurrentTime()` : Dùng thư viện “`RTCLib.h`” để lấy thời gian hiện tại và in ra màn hình OLED thời gian hiện tại bao gồm:
 - ngày/tháng/năm
 - giờ: phút: giây
- `getSlotInfo()` : Dùng để in ra màn hình OLED các thông tin:
 - Thời gian hiện tại bao gồm: ngày/tháng/năm, giờ:phút:giây.
 - Slot hiện tại đang là slot mấy, giờ giải lao của slot mấy, hoặc chưa đến thời điểm bắt đầu slot 1, hoặc đã qua thời điểm kết thúc slot cuối cùng (hoặc qua `endTime - 17:30`)
 - Chưa đến slot 1, in ra “`Day not started yet.`”
 - Đã hết slot cuối hoặc hết ngày (sau 17:30), in ra “`End day.`”
 - Đang trong slot, in ra “`Current slot: ...`” (đang slot mấy) và “`End slot: ...`” (thời điểm kết thúc slot).
 - Đang trong break, in ra “`Break time ...`” (đang break mấy) và “`End break: ...`” (thời điểm kết thúc break).
 - Thời gian kết thúc slot học hiện tại, hoặc kết thúc giờ giải lao hiện tại.
- `calculateCurrentSlot()` : Dùng để tính thời gian hiện tại đang thuộc slot mấy hoặc giờ giải lao của slot mấy.
- `setDefaultSlot()` : Dùng để thiết lập thời gian các tiết học là mặc định dành cho trường Đại học FPT TP. HCM, bao gồm:
 - Mỗi ngày có 6 slot, mỗi slot kéo dài 1 giờ 30 phút.
 - Giữa mỗi slot sẽ nghỉ 15 phút (giải lao ngắn).
 - Sau slot 3 sẽ là một thời gian nghỉ dài hơn với 30 phút.
- `isOfficeTime()` : Kiểm tra thời gian có trong giờ hành chính 7:00 - 17:30 không:
 - Nếu thời gian hiện tại là trước 7:00 thì `return -1;`

- Nếu thời gian nằm trong khoảng từ 7:00 đến 17:30 thì `return 0;`
- Nếu thời gian là sau 17:30 thì `return 1;`
- `setCustomSlot()` : Điều chỉnh thông tin reng chuông trong ngày hôm đó:
 - Nhập số slot có trong ngày, hoặc số slot còn lại tính từ thời điểm bắt đầu nhập:
 - Nếu nhập trước 7:00 thì nhập số slot có hôm đó, màn hình sẽ in ra câu hỏi input: "Enter number of slots today."
 - Nếu nhập sau 7:00 thì nhập số slot còn lại trong ngày tính từ thời điểm nhập, màn hình sẽ in ra: "Enter number of slots left today."
 - Nếu nhập sau 17:30 thì nhập số slot của ngày hôm sau, màn hình in ra: "Enter number of slots tomorrow."
 - Nhập thời lượng 1 slot, thời lượng giải lao ngắn (giải lao sau mỗi tiết học), thời lượng giải lao dài (tính bằng phút).
 - Sau slot bao nhiêu thì đến thời gian giải lao dài.
 - Sau đó hàm này sẽ dựa vào các thông tin input bên trên và tính ra slot hiện tại là slot thứ bao nhiêu nếu bắt đầu tính từ startTime (hay 7:00), và gán vào currentSlot. Sau đó tổng số slot có trong ngày sẽ được tính và gán vào numSlot.
 - Nếu tổng thời lượng của tất cả slot, thời gian nghỉ,... có thời điểm kết thúc vượt quá endTime (17:30) thì thông báo không hợp lệ. Lúc đó sẽ hiện ra 1 menu với các lựa chọn: `1. Re-enter` hoặc `2. Exit`

Ngoài ra còn các hàm và tính năng:

- `displayMenu()` : hiện menu với 3 lựa chọn:
 - Cài đặt mặc định 6 slot (`1. Set default slot`)
 - Tự điều chỉnh số slot (`2. Custom your slot`)
 - Thoát (`3. Exit`)
- `loop()` : hàm lặp của Arduino
 - Khi bắt đầu hoặc reset, OLED sẽ hiện ra thời gian hiện tại.
 - Khi bắt đầu, reset, hoặc OLED đang hiện thông tin các slot, break, nhấn phím * bất kỳ lúc nào để mở menu và set up các lựa chọn (1-3).
- `ringBellPerSlot()` : cho chuông reng vào thời điểm cần thiết.
- Nút bấm cho chuông reng bất kỳ lúc nào (BUTTON).

2. Flowchart


```

long currentSlot = 0;
int shortBreak, longBreak;

// startTime = thời gian bắt đầu ngày (bắt đầu slot 1) (7:00)
// endTime = thời gian hết ngày (17:30)
const long startTime = 7 * 3600;
const long endTime = (long)17 * 3600 + 30 * 60;

long slotInfo[200][2] {}; // [i][]: là slot thứ i, [i][0] là slot thứ i bắt đầu lúc nào, [i][1] là slot thứ
i kết thúc lúc nào, thang đo second, đánh số từ 1

int TIME_BELL_RING = 2000; //Chuông sẽ reng trong 2s

RTC_DS1307 rtc;

void setup()
{
    rtc.begin();
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);

    // setup the position 13 for bell output
    pinMode(bellPosition, OUTPUT);
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE); // Normal 1:1 pixel scale // Draw white text
    display.setCursor(0, 0); // Start at top-left corner
}

// Hàm lấy input từ Keypad, gọi hàm và trả về giá trị là input, kiểu dữ liệu integer
// Nếu không nhập gì mà nhấn # thì sẽ không đi tiếp mà cứ bắt nhập, đến khi nào có dữ liệu mới thôi
int getKeypad(String msg = "Enter number: ") {
    display.print(msg);
    display.display();
    char inputKey = keypad.getKey();
    String str = "";
    while (str == "") {
        while (inputKey != '#' && inputKey != '*') {
            if ((int)keypad.getState() == PRESSED) {
                str = str + String(inputKey);
                display.print(String(inputKey));
                display.display();
            }
            inputKey = keypad.getKey();
        }
        if (str == "") inputKey = keypad.getKey();
    }
    display.println();
    return str.toInt();
}

// Hàm cho chuông reng
void ringBell()
{
    digitalWrite(bellPosition, HIGH);
    delay(TIME_BELL_RING);
    digitalWrite(bellPosition, LOW);
}

// Hàm output OLED với 2 dòng output
void turnOnOLED(String msg1, String msg2)
{
    display.println(msg1);
    display.println(msg2);
}

// Hàm output OLED với 1 dòng output
void turnOnOLED(String msg)
{
    display.println(msg);
}

```

```

// Tắt OLED
void turnOffOLED()
{
    display.clearDisplay();
}

// Hàm lấy thời gian thực sử dụng RTCLib.h
void getCurrentTime()
{
    DateTime currentTime = rtc.now();
    int day = currentTime.day();
    int month = currentTime.month();
    int year = currentTime.year();

    String msg1 = "Date: " + String(day);
    msg1 += "/" + String(month);
    msg1 += "/" + String(year);

    int hour = currentTime.hour();
    int minute = currentTime.minute();
    int second = currentTime.second();

    String msg2 = "Time: " + String(hour);
    msg2 += ":" + String(minute);
    msg2 += ":" + String(second);

    display.setCursor(0, 0);

    display.println("Current time:");
    turnOnOLED(msg1, msg2);
}

// In ra thông tin thời gian hiện tại bao gồm
// Thời gian hiện tại
// Slot bao nhiêu hoặc end day hoặc break time
// Kết thúc slot là bao nhiêu, kết thúc break time là bao nhiêu
// Nếu numSlot == 0 thì return
// Nếu thời gian hiện tại là trước startTime thì hiện là Ngày chưa bắt đầu
// Nếu thời gian hiện tại là sau endTime thì hiện là Hết ngày
void getSlotInfo()
{
    turnOffOLED();
    getCurrentTime();

    if (numSlot == 0) {
        return;
    }

    DateTime currentTime = rtc.now();

    long currentSeconds = (long)(currentTime.hour()) * 3600;
    currentSeconds += currentTime.minute() * 60;
    currentSeconds += currentTime.second();

    if (currentSlot != 0 && currentSeconds >= slotInfo[currentSlot][0] && currentSeconds <= slotInfo[currentSlot][1]) {
        String msgSlot = "Current slot: ";
        msgSlot += String(currentSlot);

        long remain = 0;
        int hour = slotInfo[currentSlot][1] / 3600;
        remain = slotInfo[currentSlot][1] % 3600;
        int minute = remain / 60;
        int second = remain % 60;
        String msgEndTime = String(hour);
        msgEndTime += ":" + String(minute) + ":" + String(second);

        display.println(msgSlot);
        display.print("End slot: ");
        display.println(msgEndTime);
    }
}

```

```

    }
    else {
        if (currentSlot >= numSlot) {
            display.println("End day.");
        }
        else if (currentSeconds < startTime)
            display.println("Day not started yet.");
        else {
            long remain = 0;
            int hour = slotInfo[currentSlot + 1][0] / 3600;
            remain = slotInfo[currentSlot + 1][0] % 3600;
            int minute = remain / 60;
            int second = remain % 60;
            String msgEndTime = String(hour);
            msgEndTime += ":" + String(minute) + ":" + String(second);

            display.print("Break time ");
            display.println(currentSlot);
            display.print("End break:");
            display.println(msgEndTime);
        }
    }
}

// Tính hiện tại slot này là slot mấy -> lưu vào biến currentSlot
void calculateCurrentSlot()
{
    DateTime currentTime = rtc.now();

    long currentSeconds = (long)(currentTime.hour()) * 3600;
    currentSeconds += currentTime.minute() * 60;
    currentSeconds += currentTime.second();

    for (int i = 1; i <= numSlot; i++)
    {
        if (currentSeconds >= slotInfo[i][0] && currentSeconds <= slotInfo[i][1]) {
            currentSlot = i;
            return;
        }
        if (currentSeconds < slotInfo[i][0] ) {
            currentSlot = i - 1;
            return;
        }
    }
}

currentSlot = numSlot;
}

//Set số lượng slot mặc định
//Gồm 6 slot mỗi slot kéo dài 1h30ph
//Thời gian nghỉ ngắn giữa mỗi slot là 15ph
//Sau slot 3 là có 1 thời gian nghỉ dài kéo dài 30ph
void setDefaultSlot()
{
    numSlot = 6;
    long startSlot = startTime;
    const int slotDuration = 3600 + 30 * 60;

    shortBreak = 15 * 60;
    longBreak = 30 * 60;

    noSlotB4LongBreak = 3;

    for (int i = 1; i <= numSlot; i = i + 1)
    {
        slotInfo[i][0] = startSlot;
        slotInfo[i][1] = startSlot + slotDuration;

        if (i == noSlotB4LongBreak) {
            startSlot += slotDuration;
        }
    }
}

```

```

        startSlot += longBreak;
    }
    else {
        startSlot += slotDuration;
        startSlot += shortBreak;
    }
}

calculateCurrentSlot();
}

// Kiểm tra thời gian có trong giờ hành chính 7:00 - 17:30 không
// Nếu thời gian < 7:00 thì return -1
// Nếu thời gian nằm trong khoảng 7:00 đến 17:30 thì return 0
// Nếu thời gian > 17:30 thì return 1
int isOfficeTime(long currentSeconds) {
    if (currentSeconds >= startTime && currentSeconds <= endTime)
        return 0;
    else if (currentSeconds > endTime) return 1;
    else return -1;
}

// custom slot
// Nhập các thông tin: nếu trước 7:00 thì nhập số slot có hôm đó,
// nếu sau 7:00 thì nhập số slot còn lại trong ngày
// sau 17:30 thì nhập số slot ngày hôm sau
// Thời lượng 1 slot, break ngắn, break dài
// Bao nhiêu slot nữa thì break dài
// tất cả đều tính theo giây
void setCustomSlot() {
    DateTime currentTime = rtc.now();

    currentSlot = 0;
    long currentSeconds = (long)(currentTime.hour()) * 3600;
    currentSeconds += currentTime.minute() * 60;
    currentSeconds += currentTime.second();

    long slotDuration = 0, shortBreak = 0, longBreak = 0, numSlotLeft = 0;
    long startSlot = startTime;
    boolean cont;

    do {
        cont = false;
        display.clearDisplay();
        display.setCursor(0, 0);
        numSlot = 0;
        if (currentSeconds < startTime) {
            display.println("Enter number of slotstoday.");
            numSlot = getKeypad();
        }
        else if (currentSeconds > endTime) {
            display.println("Enter number of slotstomorrow.");
            numSlot = getKeypad();
        }
        else {
            display.println("Enter number of slotsleft today.");
            numSlotLeft = getKeypad();
        }
    }

    display.println("Enter slot duration.");
    slotDuration = getKeypad("Enter minute: ") * 60;

    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Short break duration.");
    shortBreak = getKeypad("Enter minute: ") * 60;
    display.println("Long break duration.");
    longBreak = getKeypad("Enter minute: ") * 60;
    display.println("Long break after slot: ");
    noSlotB4LongBreak = getKeypad();
}

```

```

    long timeUntilLongBreak = noSlotB4LongBreak * slotDuration + (noSlotB4LongBreak - 1) * shortBreak;

    // tính slot hiện tại là slot mấy
    // chia ra 2 trường hợp, trước longBreak và sau longBreak
    // nếu trước startTime hoặc sau endTime thì currentSlot = 0
    //if (numSlot == 0){
    if ((currentSeconds - startTime) <= timeUntilLongBreak && isOfficeTime(currentSeconds) == 0)
        currentSlot = (currentSeconds - startTime) / (slotDuration + shortBreak) + 1;
    else if (isOfficeTime(currentSeconds) == 0)
        currentSlot = (currentSeconds - startTime - longBreak + shortBreak) / (slotDuration + shortBreak) +
1;
    else currentSlot = 0; //}

    // tính tổng số slot trong ngày
    numSlot = numSlot + currentSlot + numSlotLeft;

    int choice;

    // Nếu số slot custom có thời gian kết thúc vượt quá endTime thì nhập lại hoặc thoát
    if (isOfficeTime(numSlot * (slotDuration + shortBreak) - shortBreak + longBreak + startTime) == 1) {
        do {
            display.clearDisplay();
            display.setCursor(0, 0);
            display.println("Endtime exceeds 17:30");
            display.println("1. Re-enter");
            display.println("2. Exit");
            choice = getKeypad();
            if (choice == 1)
                cont = true;
            else if (choice == 2) numSlot = 0;
            else {
                display.clearDisplay();
                display.setCursor(0, 0);
                display.println("Please re-enter 1-2");
                display.display();
                delay(1000);
            }
        } while (choice != 1 && choice != 2);
    }
    } while (cont);
    for ( int i = 1; i <= numSlot; i++) {
        slotInfo[i][0] = startSlot;
        slotInfo[i][1] = startSlot + slotDuration;

        if (i == noSlotB4LongBreak) {
            startSlot += slotDuration;
            startSlot += longBreak;
        }
        else {
            startSlot += slotDuration;
            startSlot += shortBreak;
        }
    }
    calculateCurrentSlot();
}

//Dựa vào thông tin của slotInfo để reng chuông
//Chuông sẽ reng vào đầu slot và cuối mỗi slot
void ringBellPerSlot()
{
    DateTime currentTime = rtc.now();

    long currentSeconds = (long)(currentTime.hour()) * 3600;
    currentSeconds += currentTime.minute() * 60;
    currentSeconds += currentTime.second();

    for (int i = currentSlot; i <= numSlot; i++)
    {
        if (currentSeconds == slotInfo[i][0]) {
            String msg = "Start slot ";
            msg += String(i);

```

```

        turnOnOLED(msg);
        currentSlot = i;
        ringBell();
    }

    if (currentSeconds == slotInfo[i][1]) {
        String msg = "End slot ";
        msg += String(i);
        turnOnOLED(msg);
        currentSlot = i;
        ringBell();
    }
}

//Hiện menu
void displayMenu()
{
    int choice = 0, cont = 0;
    do {
        turnOffOLED();
        display.setCursor(0, 0);
        display.println("1. Set default slot");
        display.println("2. Custom your slot");
        display.println("3. Exit");
        //display.display();

        choice = getKeyPad("Your choice: ");
        switch (choice) {
            case 1:
                setDefaultSlot();
                return;
            case 2:
                setCustomSlot();
                return;
            case 3:
                return;
            // nhập số khác thì re-enter
            default:
                display.clearDisplay();
                display.setCursor(0, 0);
                turnOnOLED("Please re-enter 1-3.");
                display.display();
                delay(1000);
        }
    } while (choice != 1 && choice != 2 && choice != 3);
}

void loop()
{
    getSlotInfo();

    char option = '0';
    option = keypad.getKey();
    if ((int)keypad.getState() == PRESSED && option == '*') {
        displayMenu();
    }
    ringBellPerSlot();

    display.println("Press * to set up");
    display.display();
}

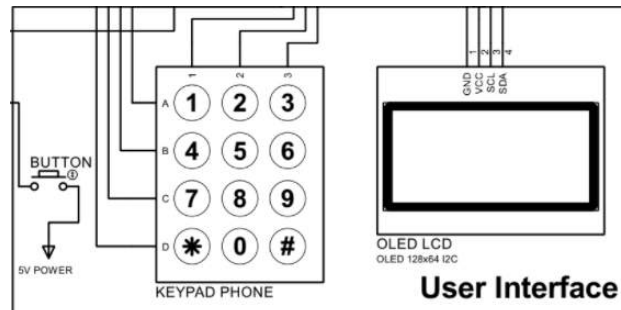
```

IV. Lắp đặt và chạy thử

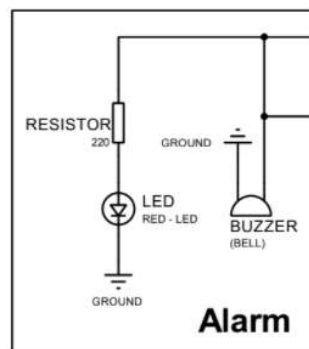
1. Thiết kế (hướng dẫn lắp đặt)

Thiết bị gồm có 3 phần:

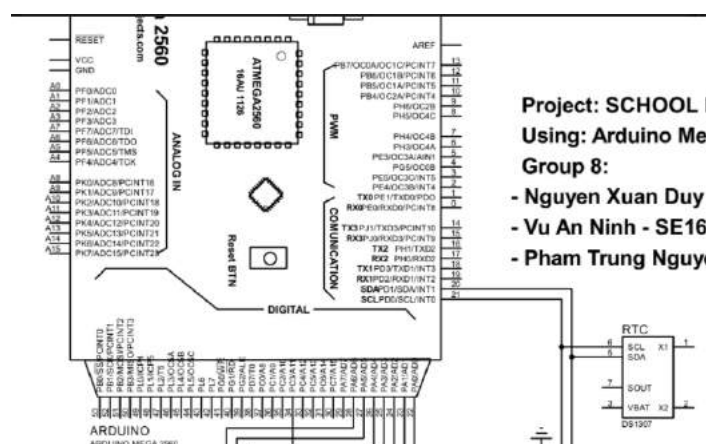
- **User Interface:** Giao diện người dùng, nơi người dùng thao tác với thiết bị thông qua các hoạt động: đọc thông tin hiển thị trên màn hình OLED, nhập thông tin vào keypad và nhấn chuông rung bất kỳ lúc nào. Nên lắp nơi dễ thao tác và quản lí.



- **Alarm:** Bộ phận chuông. Bao gồm chuông báo và đèn led báo hiệu. Lắp đặt gần khu phòng học. Có thể đấu nối để lắp thêm chuông và thêm đèn.



- **Phần cứng back-end:** Bao gồm Arduino Mega và thiết bị DS1307 lấy thời gian thực. Đây là vi xử lí code, điều khiển thiết bị. Lắp đặt nơi an toàn, không ẩm ướt hoặc ánh nắng trực tiếp (tránh hư thiết bị), hoặc tích hợp vào một chỗ với User Interface.



2. Lỗi phát sinh có thể có

- Hư các thiết bị phần cứng: Hư thiết bị nào thay thiết bị đó. Tuy nhiên nếu bộ phận hư là do quá tải điện áp, cần kiểm tra cầu chì và các thiết bị khác để chắc rằng không hư dây chuyền.
- Thời gian thực bị lệch: hạn chế của thiết bị phần cứng. Cần reset và thiết lập lại để lấy đúng thời gian thực.
- Đang sử dụng mà tắt đi bật lại: cần thiết lập lại thiết bị từ đầu.

3. Các bộ thử

- Sai số cho phép: ± 3 phút
- Tổng điểm thử nghiệm tối đa là 10.

STT	Hiển thị menu, nhập lựa chọn (1)	Kiểm tra tính hợp lệ của input (2 lần) (2)	Thời gian thử có sai số cho phép (1)	Thời gian thử (màn hình có hiển thị) (1)	Slot hiện tại (màn hình có hiển thị) (1)	Slot hiện tại (màn hình có hiển thị) (1)	Thời gian kết thúc slot có sai số cho phép (1)	Thời gian kết thúc slot (màn hình có hiển thị) (1)	Chương trình (1)	Đánh giá thử nghiệm (TỔNG 10)
1	Mặc định 10:15	2	1	1	1	1	1	1	1	Thành công 10/10
2	Custom 9:22	2	0	-1	1	1	1	1	1	Thành công 9/10
3	Custom 20:15	2	1	1	1	1	1	1	1	Thành công 10/10
4	Custom 6:45	2	1	1	1	1	1	1	1	Thành công 10/10
5	Custom 12:12	2	1	1	1	1	1	1	1	Thành công 10/10

4. Mong đợi

- Khi được lắp đặt sử dụng trong thực tế hoặc khi có điều kiện hơn về trang thiết bị, có thể sử dụng bàn phím input khác với tính năng đa dạng hơn (xóa, nhiều lựa chọn nhập).
- Tùy theo yêu cầu cụ thể của người dùng có thể thay đổi quy trình nhập dữ liệu, cách vận hành của chương trình tự động (thay đổi code).
- Khi sử dụng dành cho thương mại, hoặc phiên bản cải tiến, có thể tích hợp các ứng dụng dành cho điện thoại, máy tính (app), tích hợp nhiều tính năng hơn (input,...), kết nối với phần mềm quản lý thời gian biểu (FAP) của trường.
- Thiết kế của vỏ bao ngoài đảm bảo được độ bền và tránh được va đập nhẹ cho thiết bị bên trong chương trình. Nơi lắp đặt tuân thủ các yêu cầu được ghi ở phần IV (Lắp đặt và chạy thử), như lắp đặt nơi khô ráo, tránh ẩm ướt,...

- Khi được nâng cấp trang thiết bị đa dạng hơn, nên có thêm một phần cứng đóng vai trò như bộ nhớ lưu lại thông tin đã nhập ngày hôm trước.

V. Tổng kết

- Nhìn chung, thiết bị **Chuông báo giờ tự động trong trường học** của nhóm 5 đã đạt được các yêu cầu của dự án như lấy được thời gian thực, người dùng có thể tùy chỉnh được các dữ liệu đầu vào, chuông có thể reo bất kỳ lúc nào,...
- Thiết bị của nhóm cũng có thêm các **cải tiến**, những tính năng không có trong yêu cầu như kiểm tra định dạng input và cảnh báo, cho người dùng nhập lại nếu không đúng định dạng (kiểm tra dữ liệu đầu vào), sử dụng màn hình OLED với độ hiển thị tối ưu, sử dụng được Arduino Mega với công suất tối ưu hơn, thích hợp hơn đối với thiết bị dự án, và tích hợp nhiều lựa chọn dành cho người dùng như cài đặt chế độ mặc định, tùy chỉnh thời lượng tiết học,...
- Tuy nhiên vẫn có vài **hạn chế** về mặt mô phỏng như khả năng mô phỏng còn thấp (phụ thuộc vào laptop) dẫn đến thời gian thực bị lệch đôi chút, chưa có khả năng xóa khi nhập input (hạn chế của OLED), chưa thực sự bao quát được các rủi ro. Dự án rất cần sự góp ý và giúp đỡ từ người dùng khi sử dụng trong các tình huống thực tế và từ thầy hướng dẫn.

VI. Phụ lục

1. Link các thành phần khác

- **Flowchart** vận hành thiết bị: <https://drive.google.com/file/d/15NNpFZH5wFkQPjMuZyhhBE-Q8ReDZN02/view?usp=sharing>
- Link file **mô phỏng dự án bằng Proteus**:
<https://drive.google.com/file/d/1uCrVzpFVRy0LGhqVoBVWaSBxAp8ZPvkx/view?usp=sharing>

>>> **Lưu ý:** Tải về máy và mở file bằng phần mềm Proteus. Cần sử dụng phiên bản mới nhất (8.13 Pro) để mở được file.

- Link folder chứa **code Arduino**: <https://drive.google.com/drive/folders/1q-mNfM8Jwj0utQFiGFkhN3mUsyEOnui4?usp=sharing>

>>> **Lưu ý:** Đây là file **.ino** (mở bằng IDE Arduino). Để code compile được cần tải đầy đủ các thư viện được đính kèm ở dưới đây (các thư viện hỗ trợ xử lý chuỗi, thao tác keypad, OLED, RTC). Ngoài ra để chạy được code thì cần chỉnh IDE Arduino về cài đặt Arduino Mega.

- Link folder chứa đầy đủ file **.ZIP** các thư viện cần cài đặt (dành cho Arduino):
https://drive.google.com/drive/folders/1OPQsCNPi7yr1SriMSHNAXBtWsNMv_XBU?usp=sharing

>>> **Lưu ý:** Hướng dẫn cách cài đặt thư viện: Sketch → Include Library → Add .ZIP Library ... → chọn file .ZIP của các thư viện trong folder trên.

- Link **Schematic Design**: (có thể trực tiếp Schema ở phần II mục 3):
https://drive.google.com/file/d/1zz9H41p6eb45Zhso_FcalRxGu8tY5E3Y/view?usp=sharing

2. Nguồn tham khảo

- a) Muhammad Ansar. (2017). **Real time clock based Automatic Home Appliances control | Real Time Clock With Timer**, truy cập ngày 27/01/2022, từ <https://www.youtube.com/watch?v=DI-pfmb4nT0&t=491s>
- b) Simple Projects. (2017). **Arduino real time clock with DS1307 - Proteus simulation**, truy cập ngày 27/01/2022, từ <https://www.youtube.com/watch?v=TIS14GBBd4c&t=13s>
- c) Engr Fahad. (2020). **Arduino Oled i2c Display 128×64 with examples, Wiring, and Libraries issues solved**, truy cập ngày 28/01/2022, từ https://www.electronicclinic.com/arduino-oled-i2c-display-128x64-with-examples-wiring-and-libraries-issues-solved/#google_vignette
- d) Christopher Andrews. (2015). **Arduino Keypad Library**, truy cập ngày 28/01/2022, từ <https://playground.arduino.cc/Code/Keypad/>
- e) Satyam Singh. (2016). **How to simulate 3x4 Keypad with arduino in proteus**, truy cập 28/01/2022, từ https://www.youtube.com/watch?v=-_xB_ZkJBXU

GROUP 5 XIN CHÂN THÀNH CẢM ƠN MỌI NGƯỜI VÀ THẦY ĐÃ ĐỌC BÁO CÁO

-Hết-