**// Arduino**

**#include** "Arduino.h"

**#include** "Screen.h"

**#include** "SimpleTimer.h"

**#include** "TimeLib.h"

**#include** "JC\_Button.h"

**#include** "BluetoothModule.h"

**#include** "PedometerModule.h"

**#include** "PulseSensor.h"

//--------------------------------------------------

SimpleTimer mainTimer; **// Tạo 1 object timer**

Screen scr; //SCL SDA **// Tạo 1 object màn hình**

Button button(6); //Pin connect to Button VCC **// Tạo 1 object button kết nối pin 6**

BluetoothModule bluetoothModule(7, 8); **// Tạo 1 object bluetooth nối 2 chân 7,8 -> RX TX**

PedometerModule pedometerModule(A1, A2, A3); **// Tạo 1 object đếm bước chân sử dụng 3 chân A1, A2, A3**

PulseSensor pulseSensor(5, A0); **// Tạo 1 object đo nhịp tim, 5 nối vào nguồn, A0 đọc tín hiệu**

**int** mainTimerId; **// Lưu ID của chu kì hoạt động chính**

**int** pedoTimerId; **// Lưu ID của chu kì hoạt động cảm biến gia tốc**

**int** tempTimerId; **// Lưu ID của chu kì tạm (khi chạy đo nhịp tim)**

**void** runScreen() { **// Hàm chạy màn hình**

scr.run();

}

**void** changeScreen() { **// Thay đổi chu kì hoạt động dựa trên màn hình đang dùng**

**switch** (scr.getShowingScreenId()) {

**case** 0:

mainTimer.enable(mainTimerId);

break;

**case** 1:

mainTimer.disable(mainTimerId);

mainTimer.enable(tempTimerId);

break;

**case** 2:

mainTimer.disable(tempTimerId);

mainTimer.enable(mainTimerId);

break;

**case** 3:

default:

break;

}

scr.changeToNextScreen();

scr.run();

}

**void** pedoRun() { **// Chạy cảm biến bước chân**

pedometerModule.run();

scr.setStepCount(pedometerModule.getStepCount()); **// Đồng bộ bộ đếm bước chân của màn hình và cảm biến**

}

**void** bluetoothRun() { **// Hàm giải mã lệnh bluetooth**

**String** inputStr = bluetoothModule.getBluetoothText();

**String** inputArr[5] = { "", "", "", "", "" };

**int** count = 0;

**if** (inputStr.length() > 0) {

for (**int** i = 0; i < inputStr.length(); i++) {  **// Tách lệnh nhận được từ bluetooth vào mảng**

**if** (inputStr.charAt(i) == ',') {

count++;

} **else** {

inputArr[count] += inputStr.charAt(i);

}

}

**if** (inputArr[0] == "ST") { **// Nếu mã đầu là ST (Set Time)**

setTime(inputArr[1].toInt());  **// Mã 2 là thời gian thay đổi**

} **else** **if** (inputArr[0] == "GS") { **// Nếu mã đầu là GS (Get Step)**

bluetoothModule.sendText("GS," + **String**(scr.getStepCount())); **// Gửi lệnh số bước cho ĐT**

} **else** **if** (inputArr[0] == "GB") { **// Nếu mã đầu là GB (Get Beat)**

bluetoothModule.sendText("GB," + **String**(scr.getBpm())); **// Gửi lệnh nhịp tim cho ĐT**

}

}

}

**void** setup() { **// Hàm cài đặt chạy lúc khởi động arduino**

Serial.begin(9600); **// Serial băng tần 9600 bps**

bluetoothModule.setup(); **// Khởi động bluetoothModule**

scr.setup(); **// Khởi động màn hình**

pedometerModule.setup(); **// Khởi động cảm biến gia tốc**

button.begin(); **// Khởi động nút nhấn cảm ứng**

pulseSensor.setup(); **// Khởi động cảm biến nhịp tim**

scr.setPulseSen(pulseSensor); **// Truyền địa chỉ cảm biến đo nhịp tim cho màn hình**

setTime(1541109600); **// Cài đặt thời gian mặc định ban đầu**

mainTimerId = mainTimer.setInterval(1000, runScreen); **// Thiết lập chu kì hoạt động chính (T = 1s)**

pedoTimerId = mainTimer.setInterval(100, pedoRun); **// Thiết lập chu kì hoạt động cảm biến gia tốc (T = 0.1s)**

tempTimerId = mainTimer.setInterval(50, runScreen); **// Thiết lập chu kì tạm khi chạy đo nhịp tim (T = 0.05s)**

}

**void** loop() { **// Hàm loop chạy liên tục sau khi khởi động xong**

mainTimer.run(); **// Chạy bộ xử lý chu kì**

button.read(); **// Đọc nút nhấn**

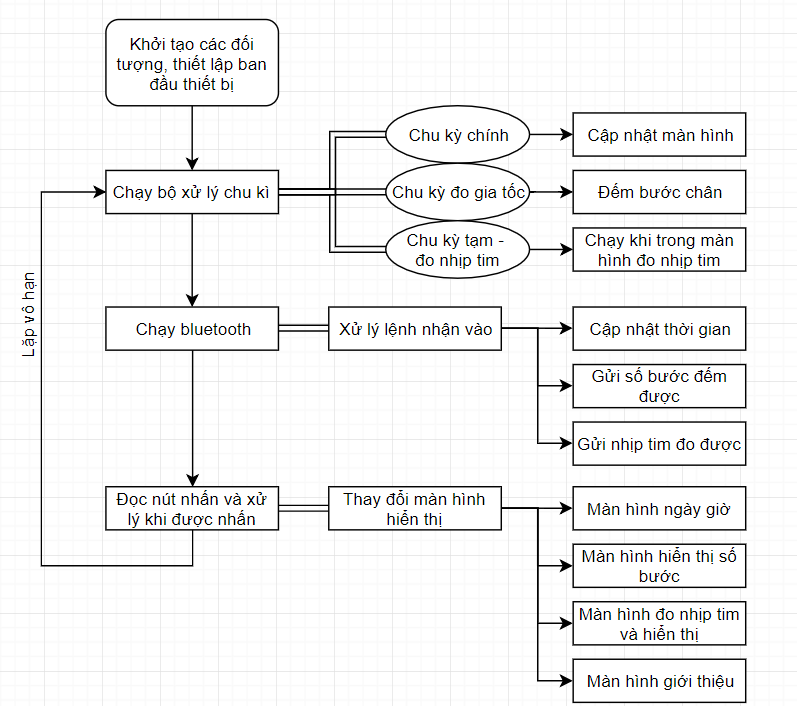
bluetoothRun(); **// Chạy bluetooth**

**if** (button.wasReleased()) { **// Nếu nút nhấn được bấm**

changeScreen(); **// Thay đổi màn hình**

}

}



**// Module Bluetooth**

**#include** "SoftwareSerial.h"

**#include** "Arduino.h"

**#include** "BluetoothModule.h"

**SoftwareSerial** bluetoothSerial(7, 8); **// Tạo object SoftwareSerial tại 2 chân RX TX là 7, 8**

**BluetoothModule**::**BluetoothModule**(**int** RX, **int** TX) { **// Hàm khởi tạo**

bluetoothSerial = **SoftwareSerial**(RX, TX);

*this*->bluetoothRecv = "";

}

**void** BluetoothModule::setup() { **// Hàm cài đặt**

bluetoothSerial.begin(9600); **// Serial chạy trên băng tần 9600 bps**

}

**String** BluetoothModule::getBluetoothText() { **// Hàm xử lý dữ liệu nhận được từ bluetooth**

**if** (bluetoothSerial.available()) {

bluetoothRecv += char(bluetoothSerial.read());

} **else** {

**if** (bluetoothRecv != "") {

**String** copy = bluetoothRecv;

*this*->bluetoothRecv = "";

return copy;

}

bluetoothRecv = "";

}

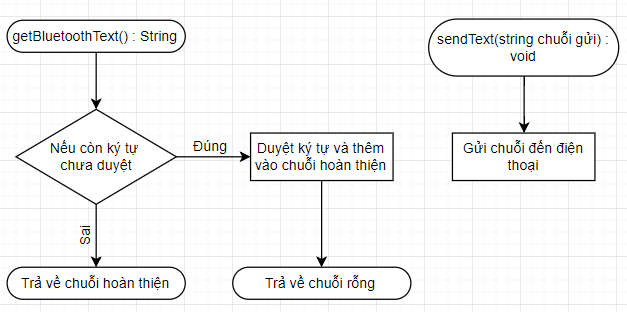
return "";

} **// Kết quả trả về là chuỗi hoàn thiện nhận được từ   
 bluetooth**

**void** BluetoothModule::sendText(**String** input) { **// Hàm gửi text từ bluetooth đến ĐT**

bluetoothSerial.write(input.c\_str());

}



**// Cảm biến nhịp tim**

**#include** "PulseSensor.h"

**#include** "Arduino.h"

PulseSensor::PulseSensor(**int** inputVCC, **int** inputSensor) { **// Hàm khởi tạo gồm chân nguồn và chân tín hiệu**

*this*->VCC\_PIN = inputVCC;

*this*->sensorPIN = inputSensor;

}

**void** PulseSensor::setup() { **// Hàm cài đặt**

pinMode(*this*->VCC\_PIN, OUTPUT);

pinMode(*this*->sensorPIN, INPUT);

digitalWrite(*this*->VCC\_PIN, LOW);

}

**int** PulseSensor::run() { **// Hàm chạy cảm biến**

digitalWrite(*this*->VCC\_PIN, HIGH); **// Khi được chạy, chân nguồn cấp 5V cho cảm nguồn cảm biến**

**int** value = *this*->readSensor(); **// Đọc giá trị khi được từ cảm biến**

**if** (value > *this*->UpperThreshold) { **// Nếu giá trị trên mức trên**

**if** (*this*->BeatComplete) { **// Nếu 1 chu kì đập hoàn thành**

*this*->BPM = millis() - *this*->LastTime; **// lấy khoảng thời gian hoàn thành 1 nhịp**

*this*->BPM = **int**(60 / (float(*this*->BPM) / 1000)); **// số nhịp trên phút bằng 60 / chu kì 1 nhịp**

*this*->BPMTiming = false; **// Thiết lập lại bộ đếm nhịp**

*this*->BeatComplete = false; **// Thiết lập lại chu kì đập**

}

**if** (*this*->BPMTiming == false) { **// Trong chu kì mới**

*this*->LastTime = millis();

*this*->BPMTiming = true;

}

}

**if** ((value < *this*->LowerThreshold) && (*this*->BPMTiming))

*this*->BeatComplete = true; **// Nếu giá trị đọc được dưới mức dưới 🡪 Hoàn thành 1 chu kì**

return value;

}

**void** PulseSensor::stop() { **// Hàm ngưng cảm biến**

digitalWrite(*this*->VCC\_PIN, LOW); **// Ngưng cấp nguồn cho cảm biến**

*this*->BPMTiming = false; **// Khởi động lại các giá trị**

*this*->BeatComplete = false;

*this*->LastTime = 0;

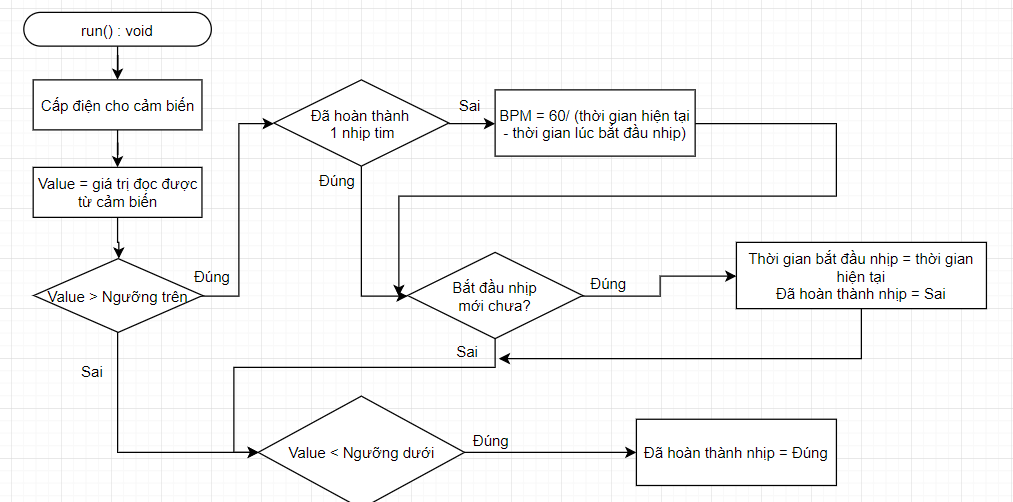
*this*->BPM = 0;

}

**int** PulseSensor::readSensor() { **// Đọc giá trị cảm biến**

return analogRead(*this*->sensorPIN);

}



**// Cảm biến gia tốc 🡪 đếm bước chân**

**#include** "PedometerModule.h"

**#include** "Arduino.h"

PedometerModule::PedometerModule(**int** in\_x\_pin, **int** in\_y\_pin, **int** in\_z\_pin) { **// Hàm khởi tạo**

*this*->x\_pin = in\_x\_pin; **// Chân đo gia tốc trục X của cảm biến**

*this*->y\_pin = in\_y\_pin; **// Chân đo gia tốc trục Y của cảm biến**

*this*->z\_pin = in\_z\_pin; **// Chân đo gia tốc trục Z của cảm biến**

**// Khởi tạo các giá trị ban đầu**

*this*->x\_val = 0;

*this*->y\_val = 0;

*this*->z\_val = 0;

*this*->dx = 0;

*this*->dy = 0;

*this*->dz = 0;

*this*->x\_last = 0;

*this*->y\_last = 0;

*this*->z\_last = 0;

*this*->threshold = 40;

*this*->flag = 0;

*this*->step = 0;

}

**void** PedometerModule::setup() { **// Hàm cài đặt cảm biến**

pinMode(x\_pin, INPUT); **// Chân x, y, z là chân đọc**

pinMode(y\_pin, INPUT);

pinMode(z\_pin, INPUT);

}

**void** PedometerModule::run() { **// Hàm chạy cảm biến**

x\_val = analogRead(x\_pin); **// Đọc giá trị từ 3 chân**

y\_val = analogRead(y\_pin);

z\_val = analogRead(z\_pin);

dx = abs(x\_val - x\_last); **// Tính toán biến thiên gia tốc**

dy = abs(y\_val - y\_last);

dz = abs(z\_val - z\_last);

**if** (dx > threshold || dy > threshold || dz > threshold) { **// Nếu biến thiên trên định mức**

**if** (flag == 0) { **// Flag = 0 : Nếu đang không trong bước chân**

*this*->step ++; **// Tăng giá trị bước chân**

flag = 1; **// Đang trong bước chân**

}

} **else** { **// Dưới mức định mức 🡪 Hoàn thành bước**

flag = 0; **// Flag = 0: Đang không trong bước chân**

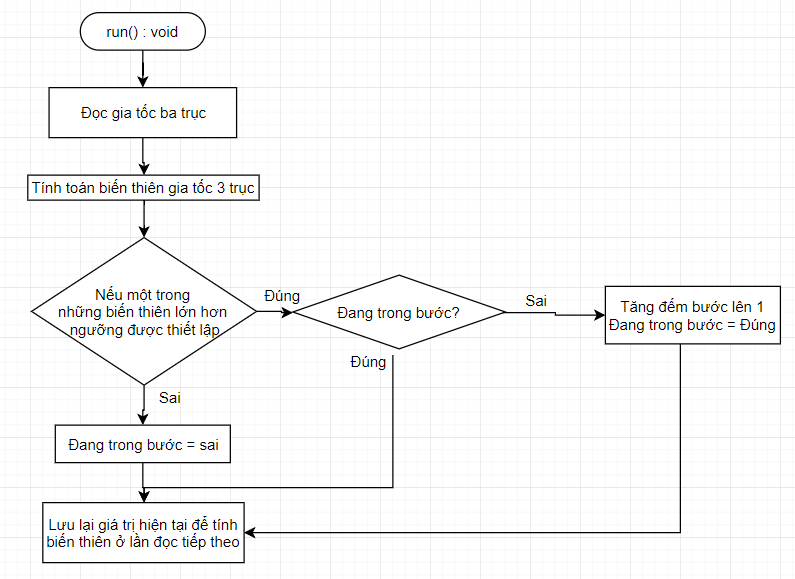
}

x\_last = x\_val; **// Lưu lại giá trị cũ để tính biến thiên tiếp theo**

y\_last = y\_val;

z\_last = z\_val;

}



**// Màn hình**

**#include** "Screen.h"

**#include** "Arduino.h"

**#include** "U8x8lib.h"

**#include** "TimeLib.h"

**#include** "PulseSensor.h"

U8X8\_SH1106\_128X64\_NONAME\_HW\_I2C u8x8(U8X8\_PIN\_NONE); **// Tạo 1 object xử lý màn hình**

Screen::Screen() { **// Hàm khởi tạo**

*this*->stepCount = 0;

*this*->bpm = 0;

}

**void** Screen::setup() { **// Hàm cài đặt**

u8x8.begin(); **// Bắt đầu chạy màn hình**

u8x8.clear(); **// Xóa màn hình**

u8x8.setFont(u8x8\_font\_chroma48medium8\_r); **// Cài đặt font sử dụng**

}

**void** Screen::showTimerScreen() { **// Hàm sử dụng khi đang trong màn hình xem thời gian**

u8x8.setCursor(0, 2); **// Đặt con trỏ tại vị trí y=0 x=2**

**String** h, m, s;

**String** d, mo, y;

**if** (hour() < 10) { **// Nếu giờ bé hơn 10 thì thêm số 0 ở trước**

h = "0" + **String**(hour());

} **else** {

h = **String**(hour());

}

**if** (minute() < 10) { **// Nếu phút bé hơn 10 thì thêm số 0 ở trước**

m = "0" + **String**(minute());

} **else** {

m = **String**(minute());

}

**if** (second() < 10) { **// Nếu giây bé hơn 10 thì thêm số 0 ở trước**

s = "0" + **String**(second());

} **else** {

s = **String**(second());

}

**if** (day() < 10) { **// Nếu ngày bé hơn 10 thì thêm số 0 ở trước**

d = "0" + **String**(day());

} **else** {

d = **String**(day());

}

**if** (month() < 10) { **// Nếu tháng bé hơn 10 thì thêm số 0 ở trước**

mo = "0" + **String**(month());

} **else** {

mo = **String**(month());

}

y = **String**(year());

**String** time = h + ":" + m + ":" + s;

**String** date = d + " " + mo + " " + y;

u8x8.setCursor(3, 1); **// Đặt con trỏ tại vị trí y=3, x=1**

u8x8.print(date); **// Hiển thị ngày tháng năm**

u8x8.draw2x2String(0, 3, time.c\_str()); **// Hiển thị giờ phút giây**

}

**void** Screen::showFootStepScreen() {

u8x8.setCursor(0, 1); **// Đặt con trỏ tại vị trí y=0, x=1**

u8x8.println("Step"); **// In giá trị số bước**

u8x8.println(*this*->stepCount);

}

**void** Screen::showHeartRateScreen() { **// Hàm sử dụng khi đang ở màn hình đo nhịp tim**

*this*->pulseSen.run(); **// Chạy cảm biến nhịp tim**

u8x8.setCursor(2, 4); **// Đặt con trỏ tại vị trí y=2, x=4**

u8x8.print("BPM: "); **// In giá trị nhịp tim trên phút đo được**

*this*->bpm = *this*->pulseSen.getBpm();

u8x8.print(*this*->bpm);

}

**void** Screen::showAboutScreen() { **// Hàm sử dụng khi đang ở màn hình giới thiệu**

u8x8.setCursor(0, 1);

u8x8.println("MyWatch");

u8x8.println("Ver 0.9b");

u8x8.println("Last mod:");

u8x8.println("23:00 - 09/11");

}

**void** Screen::run() { **// Hàm chạy màn hình**

**switch** (*this*->showingScreenId) { **// Dựa trên ID màn hình hiện tại để sử dụng hàm tương ứng**

**case** 0: **// Chạy màn hình hiển thị thời gian**

*this*->showTimerScreen();

break;

**case** 1: **// Chạy màn hình hiển thị số bước chân**

*this*->showFootStepScreen();

break;

**case** 2: **// Chạy màn hình đo nhịp tim và hiển thị nhịp tim**

*this*->showHeartRateScreen();

break;

**case** 3: **// Chạy màn hình giới thiệu (tắt đo nhịp tim)**

*this*->pulseSen.stop();

*this*->showAboutScreen();

break;

}

}

**void** Screen::changeToNextScreen() { **// Hàm sử dụng để thay đổi sang màn hình kế tiếp**

u8x8.clear(); **// Xóa màn hình hiện tại**

**if** (*this*->showingScreenId == *this*->numberOfScreen - 1) { **// Nếu đang là màn hình cuối**

*this*->showingScreenId = 0; **// Quay lại màn hình hiển thị ngày giờ**

} **else** { **// Ngược lại**

*this*->showingScreenId++; **// Hiển thị màn hình kế tiếp**

}

}

**void** Screen::setStepCount(**int** inStep) { **// Cài đặt số bước chân**

*this*->stepCount = inStep;

*this*->run();

}