Mục vẽ điểm ảnh:

app\_data->pix\_per\_rec

Mục Tổng thời gian đo:

app\_data->minutes\_for\_rec

first\_draw(): Tạo biểu đồ đầu tiên

Phân biệt

screen\_job() vs rescreen\_draw()

|  |  |  |
| --- | --- | --- |
|  | Tên | Chức năng |
| 1 | show\_screen | Hiển thị phần đầu tiên của phần mềm (Phần tiếp xúc đầu tiên) tại vì trong phần code có đoạn hiện redraw\_screen cũng như là bật nhịp tim chạy mà chưa có ghi dữ liệu |
| 2 | key\_press\_screen | Chức năng khi bấm vào nút đồng hồ |
| 3 | Menu | Hiển thị các menu chỉnh setting của phần mềm |
| 4 | first\_draw | Là lớp phủ đầu tiên lên thay thế cho lớp redraw\_screen ở dưới, cái này chỉ để thiết kế phần nhìn như chữ min, max hiện ở đâu,… và bật nhịp tim lên nhưng chưa có ghi dữ liệu |
| 5 | redraw\_screen | Là lớp phủ để hiên thị phần nhìn như min ở đâu, max hiện ở đâu (Khác với first\_draw là lúc này có lấy dữ liệu của screen\_job được rồi nên sẽ hiển thị luôn kết quả của min, max bên cạnh),… và dùng để vẽ đường hiển thị nhịp tim của biểu đồ với dữ liệu lấy chủ yếu ở screen\_job |
| 6 | draw\_scale | Vẽ cột chia số ở bên phải (20 40 60 …) |
| 7 | find\_avg | Tìm giá trị trung bình, Không cần quan tâm |
| 8 | find\_min | Tìm giá trị nhỏ nhất, Không cần quan tâm |
| 9 | find\_max | Tìm giá trị lớn nhất, Không cần quan tâm |
| 10 | update\_settings | Cập nhật lại setting |
| 11 | screen\_job\_wrapper | |
| 12 | screen\_job | Đúng như tên gọi có chữ job thì đây mới chính là cái cần vì nó bật nhịp tim lên đo rồi đưa dữ liệu vào thằng redraw\_screen để vẽ lại biểu đồ sao sao cũng như là ghi lại dữ liệu nhịp tim sao sao |
|  | dispatch\_screen | Chức năng của các bấm, quẹt trái phải |

**Nguyên lý chạy của phần mềm đo nhịp tim vẽ biểu đồ này như sau:**

Đầu tiên là phần mềm chạy phần show\_Screen, trong code phần show\_screen có mục:

set\_hrm\_mode(0); // tắt máy đo nhịp tim

fill\_screen\_bg(); // lấp đầy màn hình với màu nền (đen)

// ở đây chúng tôi thực hiện các hành động khi quay lại từ màn hình lớp phủ: khôi phục dữ liệu, v.v.

redraw\_screen(); // vẽ lại màn hình cùng với biểu đồ

set\_hrm\_mode(0x20); // bật máy theo dõi nhịp tim

set\_update\_period(1, 200);

Ở đây để ý thấy có hiện mục redraw\_screen(); nên mới vào phần mềm sẽ thấy có cấu trúc biểu đồ sẵn cũng như đồng hồ sẽ đo nhịp tim đầu tiên liền luôn do đoạn code này: set\_hrm\_mode(0x20); //

Sau đó đến mục first\_draw là vẫn chỉ có vẽ, hiển thị chữ min, max ở đâu rồi bật nhịp tim lên chớ chưa có ghi dữ liệu:

set\_backlight\_state(app\_data->backlight);

set\_bg\_color(COLOR\_BLACK); // устанавливаем цвет фона

fill\_screen\_bg(); // заполняем экран этим цветом

set\_fg\_color(COLOR\_WHITE); // устанавливаем цвет текста и линий

show\_big\_digit(3, "0", BIG\_DIGITS\_COORD\_X, BIG\_DIGITS\_COORD\_Y, 2); // отображение цифр большим шрифтом

/\*

text\_out("min:", STATS\_COORD\_X, STATS\_COORD\_Y);

text\_out("max:", STATS\_COORD\_X, STATS\_COORD\_Y + get\_text\_height());

\*/

draw\_scale(); // рисуем шкалу

show\_res\_by\_id(ICON1\_RES\_ID, ICON\_COORD\_X, ICON\_COORD\_Y); // Hình trái tim

set\_hrm\_mode(0x20); // bật máy theo dõi nhịp tim

set\_update\_period(1, 200); // включаем таймер обновления экрана

Sau đó là phần của redraw\_screen sẽ thay thế cho first\_draw tiếp tục hiển thị biểu đồ sẵn ở phần này bắt đầu có lấy dữ liệu nhịp tim từ screen\_job để vẽ đường biểu thị của biểu đồ rồi cũng như hiện luôn giá trị min max và nhịp tim hiện tại (Còn ở first\_draw là chỉ mới hiển thị thông số ban đầu như nhịp tim ban đầu là mới hiển thị là 0 vì chưa có đọc dữ liệu từ screen\_job cũng như chưa có giá trị min và max để hiện)

app\_data->curX = 0;

app\_data->curY = 176;

// вычисляем индекс первой записи на текущем экране

int i;

if (app\_data->rec\_counter > app\_data->rec\_counter\_per\_screen){

i = app\_data->rec\_counter - app\_data->rec\_counter\_per\_screen;

} else {

i = 0; // если первый экран, то индекс 0

}

app\_data->rec\_counter\_per\_screen = 0;

// в цикле заново отрисовываем график, т. к. он стирается после оверлейного экрана.

for (; i < app\_data->rec\_counter; i++){

int heartrate = app\_data->records[i];

// т. к. координата y = 0 сверху экрана, а не снизу, пересчитываем необходимые координаты

int Y = 176 - heartrate + SCALE\_OFFSET;

// в зависимости от того, повысился пульс или понизился, правильно отрисовываем вертикальную линию,

// т. к. она рисуется только в одну сторону (от меньшей координаты к большей)

if (Y < app\_data->curY){

draw\_vertical\_line(app\_data->curX, Y, app\_data->curY);

} else if (Y > app\_data->curY){

draw\_vertical\_line(app\_data->curX, app\_data->curY, Y);

}

// Здесь мы рисуем горизонтальную линию, но пришлось применить костыль со счетчиком записей, т. к. эта линия рисовалась только у первой записи.

// Оригинальный вариант закоментирован ниже. Если вы знаете, как решить эту проблему, прошу связаться со мной в личке форума MyAmazfit.ru или 4PDA. Ник написан вначале файла.

//draw\_horizontal\_line(Y, app\_data->curX, app\_data->pix\_per\_rec);

draw\_horizontal\_line(Y, app\_data->pix\_per\_rec \* app\_data->rec\_counter\_per\_screen, app\_data->pix\_per\_rec \* (app\_data->rec\_counter\_per\_screen + 1));

// заполняем структуру данными

app\_data->curX = app\_data->curX + app\_data->pix\_per\_rec; //Vẽ từng điểm ảnh dựa vào chỉ số pixel ở trên

app\_data->curY = Y;

app\_data->rec\_counter\_per\_screen = app\_data->rec\_counter\_per\_screen + 1;

repaint\_screen\_lines(0, 176); // обновление экрана

}

draw\_scale(); // рисуем шкалу

char text[10];

\_sprintf(text, "%d", app\_data->records[app\_data->rec\_counter]);

show\_big\_digit(3, text, BIG\_DIGITS\_COORD\_X, BIG\_DIGITS\_COORD\_Y, 2); // отображение цифр большим шрифтом

// выводим на экран минимальное и максимальное значения пульса

char min[10];

\_sprintf(min, "%s%d", "min:", find\_min());

text\_out(min, STATS\_COORD\_X, STATS\_COORD\_Y);

char max[10];

\_sprintf(max, "%s%d", "max:", find\_max());

text\_out(max, STATS\_COORD\_X, STATS\_COORD\_Y + get\_text\_height() + STATS\_COORD\_Y);

Nên quan trọng thứ Bin cần vẫn là screen\_job, vì dữ liệu của screen\_job bị lấy cho nhiều chỗ khác mà không có hiển thị gì nên screen\_job được định dạng là int chớ không phải là void (Void là cho không có lấy dữ liệu đọc thêm ở dưới)

Nên kiểm tra kiễm phần screen\_job này

|  |  |  |
| --- | --- | --- |
| **Type** | **Kích thước** | **Phạm vi giá trị** |
| char | 1 byte | -128 to 127 hoặc 0 to 255 |
| unsigned char | 1 byte | 0 tới 255 |
| signed char | 1 byte | -128 tới 127 |
| int | 2 or 4 bytes | -32,768 tới 32,767 hoặc -2,147,483,648 tới 2,147,483,647 |
| unsigned int | 2 or 4 bytes | 0 tới 65,535 hoặc 0 tới 4,294,967,295 |
| short | 2 bytes | -32,768 tới 32,767 |
| unsigned short | 2 bytes | 0 tới 65,535 |
| long | 8 bytes | -9223372036854775808 tới 9223372036854775807 |
| unsigned long | 8 bytes | 0 tới 18446744073709551615 |

Kiểu số thực

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Kích thước** | **Phạm vi giá trị** | **Độ chính xác** |
| float | 4 byte | 1.2E-38 to 3.4E+38 | 6 chữ số thập phân |
| double | 8 byte | 2.3E-308 to 1.7E+308 | 15 chữ số thập phân |
| long double | 10 byte | 3.4E-4932 to 1.1E+4932 | 19 chữ số thập phân |

Kiểu ký tự

|  |  |  |
| --- | --- | --- |
| **Type** | **Kích thước** | **Phạm vi giá trị** |
| char or signed char | 1 byte | -128 to 127 |
| unsigned char | 1 byte | 0 to 255 |

Kiểu void

Kiểu void có nghĩa là “không có giá trị”, nó không được dùng trong khai báo biến thông thường mà được sử dụng để chỉ định kiểu của các hàm không có giá trị trả về. Chúng ta sẽ hiểu rõ hơn về kiểu dữ liệu này trong tương lai ở các bài học về hàm, con trỏ,… Nghĩa là giá trị trả về này không sử dụng được chỉ dùng để hiển thị qua text\_out đồ thôi

Các giai đoạn còn thiếu:

Chức năng để hiển mục menu 4

Lưu ý là // Lưu ý với máy tính của Bin thì viết số nguyên là có dùng dấu phẩy ( , ) không phải dấu chấm ( . )