

Подключение семисегментного индикатора

Ниже будет показано, как подключить семисегментный индикатор к плате DE0. К другим платам он должен подключаться примерно так же.

Для подключения понадобятся плата, сам индикатор, соединительные провода и breadboard. Сначала поговорим о breadboard'e.

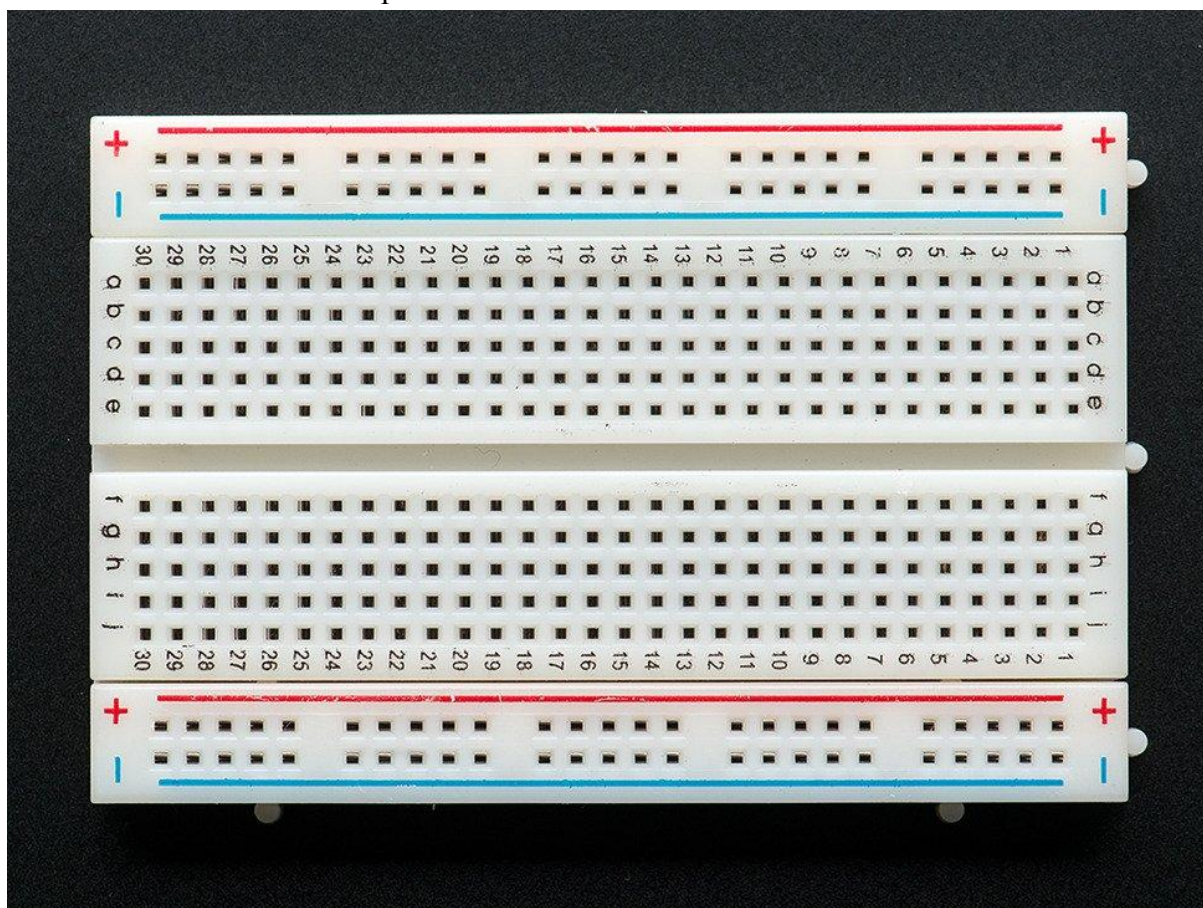


Рис. 1. Breadboard

Breadboard нужен для того, чтобы собирать на нём различные схемы. Устроена эта плата просто: двадцать пять отверстий, которые идут вдоль синей линии в самом низу платы в один ряд, являются **соединёнными друг с другом** контактами, то есть представляют собой **один** проводник. Этот факт можно как-нибудь использовать, например, соединить один из этих контактов с пином GND на плате, тем самым сделав все остальные двадцать четыре контакта аналогом GND, или же соединить какой-нибудь контакт с пином, выдающим напряжение, после чего получать необходимое напряжение из остальных контактов на этой линии.

То же справедливо и для такой же линии контактов, которая является второй снизу, а также и для двух верхних линий.

С отверстиями посередине дело обстоит иначе: они тоже соединены между собой, но более мелкими вертикальными группами по пять. Каждая такая группа лежит между красной или синей линией и продольным жёлобом, проходящим через середину платы.

Поняв, зачем нужен breadboard, перейдём к плате. В user manual по той плате, к которой подключается индикатор, найдём информацию (расположение и нумерацию) по GPIO -- группе пинов, через которые плата может обмениваться информацией с другими устройствами. На плате они выглядят как ряды тонких торчащих металлических штырьков, например, в случае DE0 это две группы по сорок таких штырьков.

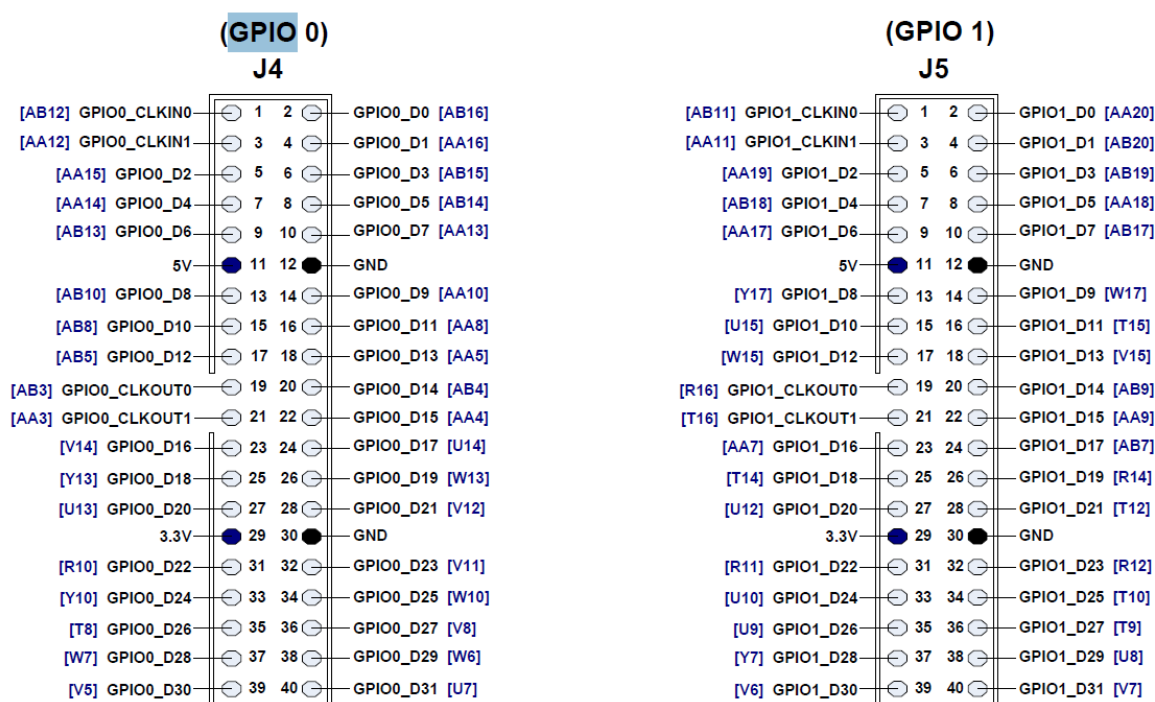


Рис. 2. Схема GPIO для DE0

Из Рисунка 2 видно, что правый верхний штырь в правой группе - это GPIO1_D0, а шестой сверху в левой колонне левой группы выдаёт напряжение в 5В. Соединяя эти штыри с чем-либо, можно выдавать информацию наружу или получать её извне.

На момент написания документа в лаборатории имелись только индикаторы BC56-12GWA, и поэтому способ подключения будет показан только для них. Официальный даташит можно найти по адресу <http://www.us.kingbright.com/images/catalog/spec/BC56-12GWA.pdf>

BC56-12GWA имеет двенадцать выводов, расположенных на тыльной стороне в два ряда, пронумерованных так, как показано на рисунке, то есть от одного до шести слева направо для нижнего ряда и от семи до двенадцати справа налево для верхнего.

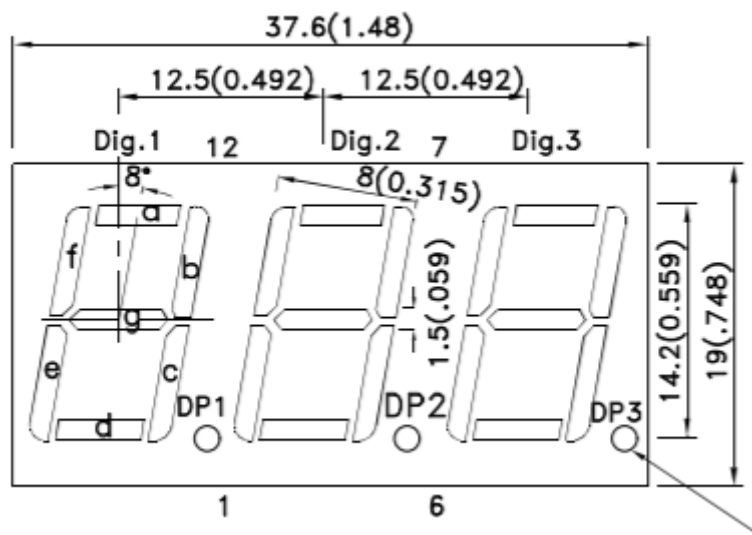


Рис. 3. BC56-12GWA

Выводы 8, 9 и 12 отвечают за зажигание нужной цифры (3, 2 и 1 соответственно, если считать слева направо) и управляются сигналом логического нуля, то есть для подачи информации на цифру номер один следует подать ноль на вывод номер 12.

Выводы 1, 2, 4, 5, 7, 10, 11 отвечают за горение сегментов, вывод 3 – за горение точки **и, в отличие от управления зажиганием цифр**, управляются сигналом логической единицы, то есть надо подавать единицу на те сегменты, которые требуется зажечь. Контакт номер 6 не используется. Подключение индикатора можно осуществлять без резисторов, напрямую. Соответствие между контактами индикатора и сегментами показано на Рисунке 4:

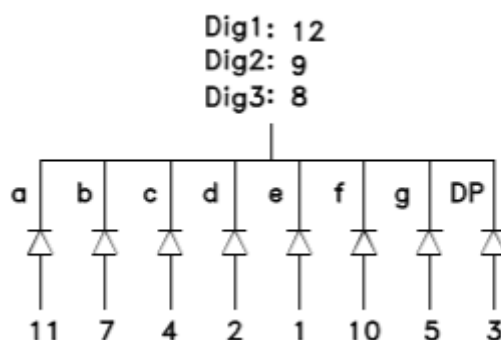


Рис. 4. Нумерация сегментов

Ясно, что индикатор не может выводить разную информацию на две или три цифры в один момент времени, поскольку это не предусмотрено его конструкцией, но можно с правильной частотой зажигать цифры с различной информацией по очереди. Это создаст видимость одновременного горения на индикаторе разных цифр.

Ниже приведен пример модуля зажигания цифр семисегментного индикатора:

```
module sm_hex_display_digit
(
    input    [ 6:0] digit1,
    input    [ 6:0] digit2,
    input    [ 6:0] digit3,
    input     clkIn,           // wire to CLOCK_50
    output reg [11:0] seven_segments
);

reg [9:0] count = 10'b0000000000;
always @(posedge clkIn)
begin
    case (count[9:8])
        2'b00: seven_segments = {1'b0, digit1[0], digit1[5], 2'b11, digit1[1],
            1'b1, digit1[6], digit1[2], 1'b0, digit1[3], digit1[4]};
        2'b01: seven_segments = {1'b1, digit2[0], digit2[5], 2'b01, digit2[1],
            1'b1, digit2[6], digit2[2], 1'b0, digit2[3], digit2[4]};
        2'b10: seven_segments = {1'b1, digit3[0], digit3[5], 2'b10, digit3[1],
            1'b1, digit3[6], digit3[2], 1'b0, digit3[3], digit3[4]};
        default: count <= count + 1'b0;
    endcase
    count <= count + 1'b1;
end

endmodule
```

Рис. 5. sm_hex_display_digit

Приведённый на Рис. 5. модуль принимает на вход тактовый импульс и семь бит для каждой цифры, указывающих, какие сегменты цифры должны гореть в данный момент, а выдаёт на выход двенадцать бит, которые управляют семисегментным индикатором. Размер регистра count, два старших бита которого отвечают за номер текущей цифры, был подобран эмпирически. Частота входного тактового импульса равна 50 МГц.

Эти семь бит, которые поступают на вход, формируются следующим модулем, который очень похож на модуль для уже присутствующего на плате семисегментного индикатора. Он принимает на вход четырёхбитное двоичное число, которое следует отобразить на одной из цифр индикатора, а выдаёт семь бит, кодирующих это число на цифре индикатора.

```

module sm_hex_display_our
(
    input      [3:0] digit,
    output reg [6:0] seven_segments
);

    always @*
        case (digit)
            'h0: seven_segments = 'b0111111; // g f e d c b a
            'h1: seven_segments = 'b0000110;
            'h2: seven_segments = 'b1011011; //  --a--
            'h3: seven_segments = 'b1001111; //  |      |
            'h4: seven_segments = 'b1100110; //  f      b
            'h5: seven_segments = 'b1101101; //  |      |
            'h6: seven_segments = 'b1111101; //  --g--
            'h7: seven_segments = 'b0000111; //  |      |
            'h8: seven_segments = 'b1111111; //  e      c
            'h9: seven_segments = 'b1100111; //  |      |
            'ha: seven_segments = 'b1110111; //  --d--
            'hb: seven_segments = 'b1111100;
            'hc: seven_segments = 'b0111001;
            'hd: seven_segments = 'b1011110;
            'he: seven_segments = 'b1111001;
            'hf: seven_segments = 'b1110001;
        endcase

endmodule

```

Рис. 6. sm_hex_display_our

Подключим модули sm_hex_display_our и sm_hex_display_digit, добавив в de0_top.v код на Рис. 7., и соединим штыри GPIO1_D1 - GPIO1_D12 проводами с соответствующими контактами индикатора. Контакт номер 6 можно ни с чем не соединять, и, конечно, можно выбрать другие пины:

```

assign GPIO1_D[12:1] = seven_segments;
wire [31:0] our_h7segment = // Что-нибудь, три младшие шестнадцатеричные
// цифры чего читатель хочет видеть на индикаторе.

wire [11:0] seven_segments;
sm_hex_display_digit sm_hex_display_digit
(
    .digit1      ( digit1      ),
    .digit2      ( digit2      ),
    .digit3      ( digit3      ),
    .clkIn       ( clkIn       ),
    .seven_segments ( seven_segments )
);

wire [ 6:0] digit3;
wire [ 6:0] digit2;
wire [ 6:0] digit1;

sm_hex_display_our digit_00 ( our_h7segment [ 3: 0] , digit3 [6:0] );
sm_hex_display_our digit_01 ( our_h7segment [ 7: 4] , digit2 [6:0] );
sm_hex_display_our digit_02 ( our_h7segment [11: 8] , digit1 [6:0] );

```

Рис. 7. Подключение модулей