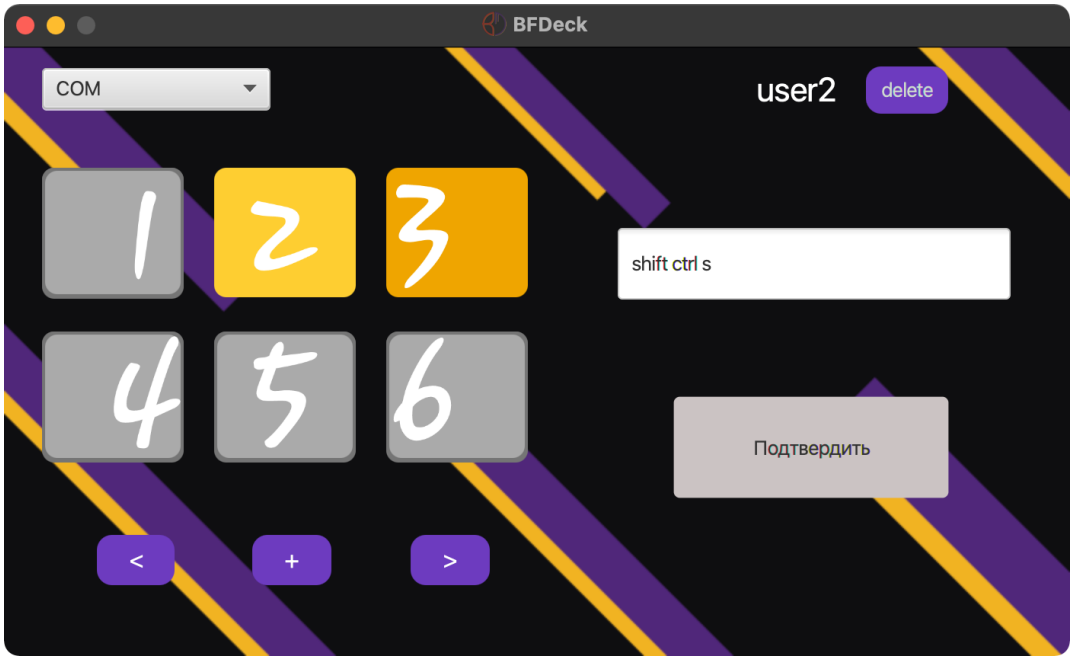
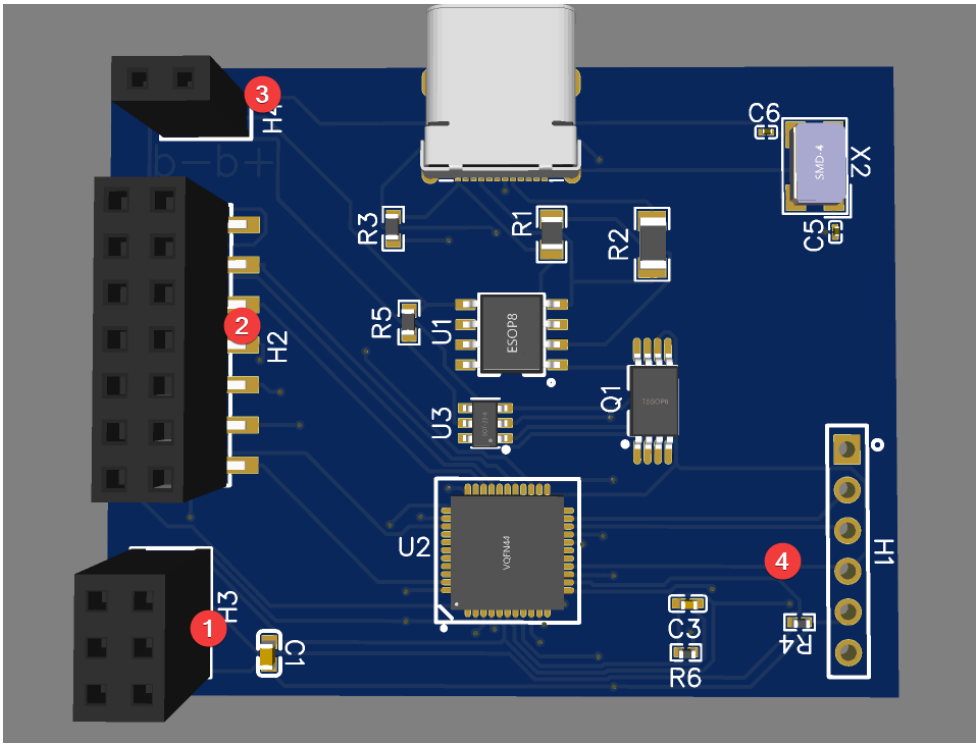
**====Вступление====**

Поскольку приложение соединяется с BFDeck через com-port, общение происходит через него. Для того, чтобы сказать, что какая-то кнопка нажата, используются команды clk1,clk2,clk3 и т.д. На картинке справа можно увидеть порядок клавиш.

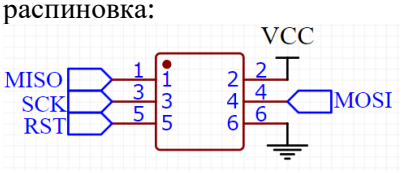



**===Схема техника===**

Рассмотрим на примере беспроводной версии, которая отличается только наличием в ней разъёма для модуля Bluetooth и разъёма для батарейки (для удобства разъёмы пронумерованы).

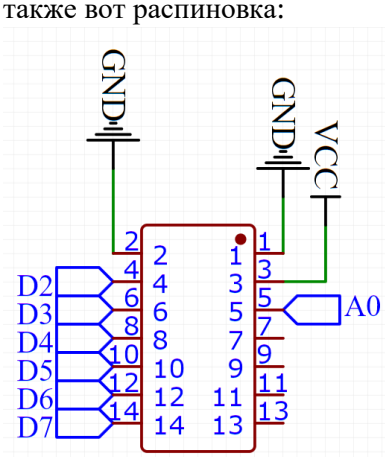



1) Это разъём для подключения прогроматора вот распиновка:




Опорная точка находится в правом нижнем углу.

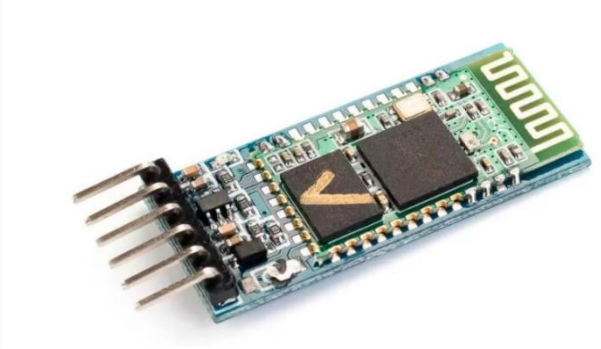
2) Это разъём для подключения кнопок и ленты также вот распиновка:




D2-D7 — пины кнопок 1-6 соответственно
A0 — пин ленты
VCC — 5V
GND —земля

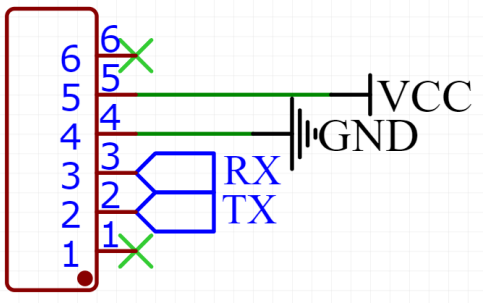
3) Распиновка разъём для аккумулятора находится на шелкографии.

4) Это разъём для подключения Bluetooth. Мы использовали данный модуль:



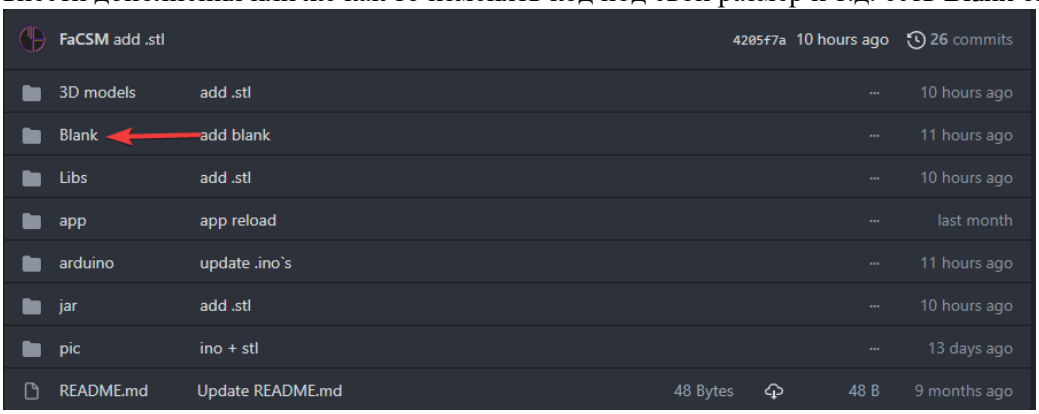
HC-05

Он подключается снизу платы пином State в разёъом помеченный точкой.Для тех кому требуется также вот распиновка разъёма:




**===Код===**

На данный момент есть готовый код для BFDeck размеро 3*2 или же 6 клавиш.Так же для тех, кто хочет внести дополнения или же как то изменять код под свой размер и т.д. есть Blank он также есть в репазитории:




Здесь вы найдете библиотеки а так же разобранный код и режими подсветки. Далее я кратко пройдусь по нему. В файле с названием BFDeck_x6_BLANK

В самом верху находится конфигурация:

```
//=======================КОНФИГУРАЦИЯ=======================//
#define backlight 1          //наличин подсветки 1-есть,0-нету
#define wireless 0           //наличие модуля bluetooth 1-есть,0-нету
#define deck_length 3        //кол-во клавиш в длинну              <------------]
#define deck_wight 2         //кол-во клавиш в ширину              <------------|
const int MAX_LEVEL = 100;   //максимальный уровень для Simon`s Says            |
const int NUM_LEDS = 6;      //это кол-во светодиодов (состоит из длинны*шарину)__]
//==========================================================//
```

Это требуетья для того чтобы пк мог узнать параметры устройства.

```
void setup() {
 Serial.begin(9600);            //инициализация serial для общения через com-port
  but1.setTickMode(AUTO);       //настройка анти-дребезга для кнопки
  but2.setTickMode(AUTO);       //настройка анти-дребезга для кнопки
  but3.setTickMode(AUTO);       //настройка анти-дребезга для кнопки
  but4.setTickMode(AUTO);       //настройка анти-дребезга для кнопки
  but5.setTickMode(AUTO);       //настройка анти-дребезга для кнопки
  but6.setTickMode(AUTO);       //настройка анти-дребезга для кнопки
  for(int i = 0; i < deck_length; i++){  //
    int u = 0;                           //
    rainm[i] = u;                        //функция для настройки радуги(требуется для коректной работы радуги на любой длинне устр-ва)
    u = u + 150;                         //
  }                                      //
}
```

Это setup тут проходит авто настройка кнопок для защиты от зачитывания “фантомных” нажатий, также активируется Serial и небольшой цикл для созданий параметров радуги.

Далее идет loop:

```
//==================двух сторонее общение МК с ПК через протакол==================//
  if (Serial.available()) {
    for (int u = 0; u < 50; u++) {
      ints[u] = 0;
    }
    char buf[255];
    for (int u = 0; u < 255; u++) {
      buf[u] = 0;
    }
    Serial.readBytesUntil('\n', buf, 255);
    Parser data(buf, ',');
    data.parseInts(ints);
   // for (int u = 0; u < 50; u++) {      //код отладки для проверки получаемых данных
   //   Serial.print(ints[u]);            //код отладки для проверки получаемых данных
   //   Serial.print(",");                //код отладки для проверки получаемых данных
   // }                                   //код отладки для проверки получаемых данных
    switch (ints[0]) {
      case 0:{
        led_mode = ints[1];
        bright_flag = false;
        break;}
      case 1:{
        periodMs = ints[1];
        break;}
      case 2:
        {bright = ints[1];
        break;}
      case 3:
        {RGB[0] = ints[1];
        RGB[1] = ints[2];
        RGB[2] = ints[3];
        break;}
      case 4:
        {int n = 1;
        for (int x = 0; x < NUM_LEDS; x++) {
          for (int y = 0; y < 3; y++) {
            color_set[x][y] = ints[n];
            n++;
          }
        }
        break;}
      case 10:
        {Serial.print(deck_length);
        Serial.print(",");
        Serial.print(deck_wight);
        Serial.print(",");
        Serial.print(backlight);
        Serial.print(",");
        Serial.println(wireless);
        break;}
    }
  };
//================================================================================//
```

Тут мы проверяем наличие новых данных и их обработка для преобразования их в команды.
Чуть позже поясню как это работает и как работает протокол.
Всё что скажу данный участок не требуется трогать если вы не хотите добавить параметры для общения.

Далее блок смена режима подсветки:

```
//==================смена режима подсветки если она пристутствует(удалить блок при не надобности)==================//
  if (now - lastTime > periodMs) {
    lastTime = now;
    switch (led_mode) {
      case 0:
        rainbow();
        break;
      case 1:
        rainbow2();
        break;
      case 2:
        temp();
        break;
      case 3:
        blinking();
        break;
      case 4:
        fill();
        break;
      case 5:
        RGBchange();
        break;
      case 10:
        Simon_loop();
        break;
    }
  };
//=============================================================================================================//
```

Если хотите добавить режим вам сюда.
Далее идёт блок проверки кнопок:

```
//=========================Код проверки нажатия кнопки(маштабировать при изменении кол-ва)=========================//
  if (but1.isClick()) {
    Serial.println("clk1");
  };
  if (but2.isClick()) {
    Serial.println("clk2");
  };
  if (but3.isClick()) {
    Serial.println("clk3");
  };
  if (but4.isClick()) {
    Serial.println("clk4");
  };
  if (but5.isClick()) {
    Serial.println("clk5");
  };
  if (but6.isClick()) {
    Serial.println("clk6");
  };
}
//=============================================================================================================//
```

Тут проверяется нажатали какая-то кнопка и передат информацию об этом ПК,тем самым способом что описан в самом верху.
На этом кончаеться основной файл,но также есть 2й led.ino в нём находятся все режими для подсветки, все режими которые вы хотите иметь надо скопировать отсюда в самый конец основного файла.

**===Протокол===**

Как же он работает? На самом деле всё не так сложно. Для пояснения есть таблица.

| | | Default`s | Existing value |
|---|---|---|---|
| 0 | режим свечения | 0 | (0-) |
| 1 | скорость | 10 | (10-2^16) |
| 2 | яркость | 255 | (0-255) |
| 3 | цвет RGB(r,g,b) | 0,0,0 | (0-255),(0-255),(0-255) |
| 4 | цвет RGB каждого светодиода | 0,0,0 | (0-255),(0-255),(0-255) |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |

0 — Это выбор режима свечения подсветки например отправив в Com-port ‘0,0” вы получите первый режим, так называемая радуга или же rainbow для того чтобы понять какой режим являеться каким номером достточно вернуться к коду и посмотреть на данный switch

```
//==================смена режима подсветки если она пристутствует(удалить блок при не надобности)==================//
  if (now - lastTime > periodMs) {
    lastTime = now;
    switch (led_mode) {
      case 0:
        rainbow();
        break;
      case 1:
        rainbow2();
        break;
      case 2:
        temp();
        break;
      case 3:
        blinking();
        break;
      case 4:
        fill();
        break;
      case 5:
        RGBchange();
        break;
      case 10:
        Simon_loop();
        break;
    }
  };
//=================================================================================================================//
```

1 — изменение скорости эффектов возможные значения от 10 до 2^16 где 10 самая быстрая а 2^16 самый медленный(хз кому надо так медленно, но это ограничения переменной)

2 — изменяет яркость подсветки где 0 выключена, а 255 максимальная яркость. Пример передачи “2,255”

3 — это задаёт цвет всех светодиодов используется в режими подсветки blink. Пример передачи “3,255,0,0” – это будет красный

4 — цвет каждого светодиода требуется для режима fill что задать цвет каждого светодиода.

10 — Он вызываеться при каждом подключении к ПК требуеться как раз чтобы пк знал какие параметры у вашей BFDeck.