```
#include <FastLED.h>
#define NUM_LEDS 99
#define DATA_PIN 6

CRGB leds[NUM_LEDS];
bool do_update = true;
int b;
char received_char;
String string_in = "";
String data_in = "";
String hex = "";

char lb = 0;
char hb = 0;
int val = 0;

int LED_to_address = 0;
int val_red = 0;
int val_green = 0;
int val_blue = 0;

int hexToValue(char H, char L){
  H = hexVal(H);
  L = hexVal(L);
  int k = H << 4;
  k += L;
  return(k);
}

char hexVal(char G){
  if(G >= 48 && G <= 57){ G -= 48; }
  if(G >= 65 && G <= 70){ G -= 55; }
  if(G >= 97 && G <=102){ G -= 87; }
  if(G >= 16){ G=0; }
  return(G);
}

void setup() {
  // put your setup code here, to run once:
 FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);

  // make sure all LEDs begin off
  for(int i=0; i < NUM_LEDS; i++){
    leds[i] = CRGB::Black;
  }

  // these are just example colours
  leds[0] = CRGB::Red;
  leds[1] = CRGB::Blue;
  leds[2] = CRGB::Green;

  Serial.begin(9600);
  Serial.println(F("Let's go!"));

}

void loop() {
  // put your main code here, to run repeatedly:

  while(Serial.available() > 0){
    received_char = Serial.read();
    if(received_char == 0x0a){
      // this is the newline character, process the
      // incoming message
      data_in = string_in;
      string_in = "";
      break;

    }else if(received_char >= 32){
      string_in += received_char;
    }

  }

  if(data_in.length() > 0){
    // parse the incoming message

    if(data_in.length()==8){


      // the first two characters should be
      // the LED number (in hex)
      hb = data_in[0];
      lb = data_in[1];
      val = hexToValue(hb, lb);

      LED_to_address = val;

      hb = data_in[2];
      lb = data_in[3];
      val = hexToValue(hb, lb);

      val_red = val;

      hb = data_in[4];
      lb = data_in[5];
      val = hexToValue(hb, lb);
      val_green = val;

      hb = data_in[6];
      lb = data_in[7];
      val = hexToValue(hb, lb);

      val_blue = val;

      leds[LED_to_address].red = val_red;
      leds[LED_to_address].green = val_green;
      leds[LED_to_address].blue = val_blue;
      do_update = true;

      // testing:
      Serial.print(F("Led "));
      Serial.print(LED_to_address);
      Serial.print(F(" set to "));
      Serial.print(val_red);
      Serial.print(F(","));
      Serial.print(val_green);
      Serial.print(F(","));
      Serial.println(val_blue);
    }

    data_in = "";
  }

  if(do_update==true){
    Serial.println(F("updating"));

    FastLED.show();
    do_update = false;
  }

}
```