

```

1  /*****
2  This is an example for our nRF52 based Bluefruit LE modules
3
4  Pick one up today in the adafruit shop!
5
6  Adafruit invests time and resources providing this open source code,
7  please support Adafruit and open-source hardware by purchasing
8  products from Adafruit!
9
10 MIT license, check LICENSE for more information
11 All text above, and the splash screen below must be included in
12 any redistribution
13 *****/
14 #include <bluefruit.h>
15
16 BLEDis bledis;
17 BLEHidAdafruit blehid;
18
19 bool hasKeyPressed = false;
20
21 void setup()
22 {
23   Serial.begin(115200);
24   while ( !Serial ) delay(10); // for nrf52840 with native usb
25
26   Serial.println("Bluefruit52 HID Keyboard Example");
27   Serial.println("-----\n");
28
29   Serial.println();
30   Serial.println("Go to your phone's Bluetooth settings to pair your device");
31   Serial.println("then open an application that accepts keyboard input");
32
33   Serial.println();
34   Serial.println("Enter the character(s) to send:");
35   Serial.println();
36
37   Bluefruit.begin();
38   Bluefruit.setTxPower(4); // Check bluefruit.h for supported values
39   Bluefruit.setName("Bluefruit52");
40
41   // Configure and Start Device Information Service
42   bledis.setManufacturer("Adafruit Industries");
43   bledis.setModel("Bluefruit Feather 52");
44   bledis.begin();
45
46   /* Start BLE HID
47   * Note: Apple requires BLE device must have min connection interval >= 20m
48   * ( The smaller the connection interval the faster we could send data).
49   * However for HID and MIDI device, Apple could accept min connection interval
50   * up to 11.25 ms. Therefore BLEHidAdafruit::begin() will try to set the min and max
51   * connection interval to 11.25 ms and 15 ms respectively for best performance.
52   */
53   blehid.begin();
54
55   // Set callback for set LED from central
56   blehid.setKeyboardLedCallback(set_keyboard_led);

```

```

57
58  /* Set connection interval (min, max) to your preferred value.
59  * Note: It is already set by BLEHidAdafruit::begin() to 11.25ms - 15ms
60  * min = 9*1.25=11.25 ms, max = 12*1.25= 15 ms
61  */
62  /* Bluefruit.Periph.setConnInterval(9, 12); */
63
64  // Set up and start advertising
65  startAdv();
66  }
67
68  void startAdv(void)
69  {
70  // Advertising packet
71  Bluefruit.Advertising.addFlags(BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE);
72  Bluefruit.Advertising.addTxPower();
73  Bluefruit.Advertising.addAppearance(BLE_APPEARANCE_HID_KEYBOARD);
74
75  // Include BLE HID service
76  Bluefruit.Advertising.addService(blehid);
77
78  // There is enough room for the dev name in the advertising packet
79  Bluefruit.Advertising.addName();
80
81  /* Start Advertising
82  * - Enable auto advertising if disconnected
83  * - Interval: fast mode = 20 ms, slow mode = 152.5 ms
84  * - Timeout for fast mode is 30 seconds
85  * - Start(timeout) with timeout = 0 will advertise forever (until connected)
86  *
87  * For recommended advertising interval
88  * https://developer.apple.com/library/content/qa/qa1931/\_index.html
89  */
90  Bluefruit.Advertising.restartOnDisconnect(true);
91  Bluefruit.Advertising.setInterval(32, 244); // in unit of 0.625 ms
92  Bluefruit.Advertising.setFastTimeout(30); // number of seconds in fast mode
93  Bluefruit.Advertising.start(0); // 0 = Don't stop advertising after n seconds
94  }
95
96  void loop()
97  {
98  // Only send KeyRelease if previously pressed to avoid sending
99  // multiple keyRelease reports (that consume memory and bandwidth)
100  if ( hasKeyPressed )
101  {
102  hasKeyPressed = false;
103  blehid.keyRelease();
104
105  // Delay a bit after a report
106  delay(5);
107  }
108
109  if (Serial.available())
110  {
111  char ch = (char) Serial.read();
112

```

```
113 // echo
114 Serial.write(ch);
115
116 blehid.keyPress(ch);
117 hasKeyPressed = true;
118
119 // Delay a bit after a report
120 delay(5);
121 }
122 }
123
124 /**
125  * Callback invoked when received Set LED from central.
126  * Must be set previously with setKeyboardLedCallback()
127  *
128  * The LED bit map is as follows: (also defined by KEYBOARD_LED_* )
129  *   Kana (4) | Compose (3) | ScrollLock (2) | CapsLock (1) | Numlock (0)
130  */
131 void set_keyboard_led(uint16_t conn_handle, uint8_t led_bitmap)
132 {
133     (void) conn_handle;
134
135     // light up Red Led if any bits is set
136     if ( led_bitmap )
137     {
138         ledOn( LED_RED );
139     }
140     else
141     {
142         ledOff( LED_RED );
143     }
144 }
```